Four-field finite element solver and sensitivities for quasi-Newtonian flows

Nathan Martin, Jerome Monnier

FOUR-FIELD FINITE ELEMENT SOLVER AND SENSITIVITIES FOR QUASI-NEWTONIAN FLOWS

N. MARTIN*† AND J. MONNIER †

Abstract. A computationally efficient finite element algorithm for power law fluid is elaborated in view of extensive direct and inverse simulations. We adopt a splitting technique to simplify the nonlinear structure of the fluids equations and derive a four-field saddle point formulation for which we prove the existence of a solution. The resolution of the corresponding variational inequalities is based on an augmented Lagrangian method and a mixed finite element discretization. The resulting iterative solver reveals to be fast and robust with low memory consumption. The time-saving provided by the algorithm compared to the standard algorithms of fixed point and Newton increases with the number of degrees of freedom and the nonlinearity of the problem. It is therefore well-suited for the solution of large problems with a great number of elements and for corresponding adjoint-based computations. Bidimensional numerical experiments are performed on two realistic situations of gravity flows: an experimental viscoplastic steady wave and a continental glacier. In the present study, results emphasize that for both cases, the modeling at bottom plays a strongly dominant role. Using surface velocitiy observations, the sensitivity analysis with respect to a spatially varying power-law exponent highlights the importance of an accurate knowledge of the rheology at high shear rate. The one on the basal sliding allows to detect the presence of a short wavelength (two times the thickness) free-slip area indetectable from surface velocities.

Key words. four-field saddle-point, augmented Lagrangian, mixed finite element, viscoplastic, adjoint method, variational sensitivities

1. Introduction. An isotropic relationship between the extra-stress tensor $\underline{\boldsymbol{S}}$ and the rate of deformation tensor $\underline{\boldsymbol{D}}$ leads to the *generalized Newtonian model*, see *e.g.* [6]. It usually expresses η as a function of the shear rate $\dot{\gamma}$ (which is itself a function of the second invariant of the rate of deformation tensor $\underline{\boldsymbol{D}}$):

$$\underline{\boldsymbol{S}} = 2\eta(\dot{\gamma})\underline{\boldsymbol{D}}(u) \tag{1}$$

We focus on a standard type of constitutive equation called *power-law* representing the viscosity η as a power of the shear-rate $\dot{\gamma}$. Hereafter we consider the classical viscoplastic Norton-Hoff law or simply *power-law* (see *e.g.* [10]). These laws occur in many mathematical models of physical process like polymeric solutions (see *e.g.* [10], [29]), mud flows (see *e.g.* [41], [15]), ice flows (see *e.g.* [14],[31]), avalanches and debris (see *e.g.* [5],[34]), lava (see *e.g.* [44],[21]), etc. Many of these flows involve gravity-driven mass movements and can thus require the modeling of a free-surface moving in time (see *e.g.* [4]).

Approximations of these flows are generally computed considering the velocity-pressure Stokes system described using finite element method. In appropriate Sobolev spaces, enhanced inf-sup condition for the power-law two-field problem and error estimates on the corresponding mixed finite element approximation are proved in [7] and improved in [45] (see also [9]). Another approach consists in deducing a nonlinear *three-field formulation* for the incompressible power-law Stokes' system based on the viscosity dependence on the second invariant of the rate of deformation tensor. For a given domain Ω, existence and uniqueness of the power-law continuous problem (treated as a limit case of the White-Metzner viscoelastic model) and of its three-field finite element approximation are given in [8] as well as error bound for homogeneous Dirichlet boundary conditions. Finer error estimates are obtained in [38] using the Cauchy stress tensor $\underline{\boldsymbol{\sigma}}$ as an unknown instead of the extra-stress tensor $\underline{\boldsymbol{S}}$. Similar results are obtained for less regular velocity, more general boundary conditions and without the assumption of an invertible constitutive relationship in [18]. It is important to notice that [38] and [18] treat with a Ladyzenskaya constitutive relationship that considers the norm of the velocity gradient instead of the rate of deformation tensor (thus, deviatoric tensor and stress tensor are not symmetrical). In terms of numerical experiments, they use the conforming finite element basis constructed in [20] based on the Raviart-Thomas finite element which provides local (at element level) conservation of momentum and mass; the nonlinear problem is solved by a fixed point algorithm.

* *Corresponding author*: nmartin@insa-toulouse.fr
† INSA, Institut de Mathématiques de Toulouse, 135 Avenue de Rangueil, 31400, Toulouse, France

We elaborate here an augmented Lagrangian type solver based on duality methods and splitting technique. The rate of deformation tensor $\underline{\boldsymbol{D}}$ is introduced as an unknown of the problem (see [25, 24]). It leads to a *four-field saddle-point formulation*. An extended Taylor-Hood discrete representation with quadratic continuous velocity, linear continuous pressure and linear discontinuous extra-stress and rate of deformation tensor is considered for the discretization. Using a duality theorem and convexity arguments, we ensure the existence of a saddle-point for this formulation in power-law index-dependent functional spaces. The incompressibility constraint is imposed in a weak sense. We elaborate a new algorithm that dramatically reduces computational time for direct solving and succeeds in a larger range of power-law exponent than classical fixed point method. In addition, the discontinuous piecewise linear discretization of the tensorial unknowns leads to a very convenient block diagonal matrix structure. This structure is exploited to obtain a very fast iterative solver with very low memory needs. Similar augmented Lagrangian type algorithm based on splitting techniques being suited for non-differentiable variational problem, it has been mainly employed for the solution of non Newtonian flows involving a yield stress (see [27]). It has been introduced to solve the p-Laplacian problem (which is a power-law type problem for Ladyzenskaya fluids) by [26] (see also [24]). To the best of our knowledge, a four-field saddle-point formulation has never been derived for the present power-law Stokes problem and similar algorithmic efficiency has not been demonstrated. The present implementation provides an iterative time (*i.e.* the time spent in the nonlinear loop) which grows with the number of degree of freeedom and the global time of computation thus converges to the time required for the solution of one Stokes system.

The second part of this work consists in building a variational sensitivity analysis tool based on an adjoint method. The adjoint model is obtained using algorithmic differentiation of the source code. The adjoint-state method provides the derivatives of a scalar function for every direction of derivation for a cost independent of the control space dimension. It is therefore an efficient method for computing local sensitivities for large-scale systems and a crucial step within an adjoint-based parameter identification process (see *e.g.* [11]). Indeed, parameters such as the basal friction, the power-law exponent or the rheological rate factor are empirical parameters that include various physical effects and the sensitivity analysis allows to weight and compare underlying physics at work. Moreover, these quantities can be hard or impossible to measure and sensitivity analysis offers perspectives to help defining experimental protocols regarding of measurements location and quality.

The present study focuses on the sensitivity analysis with respect to spatially distributed parameters, namely the basal friction coefficient β and the power-law exponent s. The continuous adjoint system associated to the power-law Stokes problem can be found in [19] for a general optimal control framework. Parameter identifications based on this adjoint model are performed in [42] in a glaciological context but they do not carry out sensitivity analyses.

As a preliminary, analytical calculations lead us to an explicit Poiseuille-like solution for the bidimensional steady uniform power-law Stokes problem with a basal sliding boundary condition. We then present sensitivity analyses, with respect to the power-law index and the basal sliding coefficient, on two realistic gravity flows: an experimental viscoplastic steady wave and a continental glacier. The explicit velocity field is used as an inflow profile. In the present case, the problem will be treated for a fixed surface, thus providing quasi-static sensitivities of the flow with respect to its input parameters. The numerical results highlight the major contribution of the basal modeling. For the small-scale experimental kaolin wave, we observe the dominant role of the rheological behavior close to the bottom (*i.e.* at high-shear rate); the large scale glacier shows a highly correlated response to the modeling of the bedrock sliding which is an empirical description adjusted by an unmeasurable quantity.

2. The Fluid Model. We present in this Section the mathematical fluid model which is the incompressible Navier-Stokes system with low Reynolds approximation combined with a power-law constitutive equation. The *Cauchy stress tensor* is expressed by:

$$\underline{\boldsymbol{\sigma}} = -p\underline{\boldsymbol{Id}} + \underline{\boldsymbol{S}} \tag{2}$$

where $\underline{\boldsymbol{S}}$ represents the deviatoric tensor and p the pressure. It is classical to characterize the rate of strain by the *shear rate* $\dot{\gamma}$ that corresponds, for a pure shear flow on an horizontal plane bathymetry, to the velocity gradient $\dfrac{\partial u_x}{\partial z}$ (in the euclidian frame (x, z) and for a corresponding velocity field $\boldsymbol{u} = (u_x, u_z)^T$). In the general case, the shear rate is a function of the second invariant of the *rate of deformation* tensor:

$$\underline{\boldsymbol{D}} = \underline{\boldsymbol{D}}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \tag{3}$$

Writing $\mathbf{D}_{(II)}$ this invariant, we obtain:

$$\dot{\gamma} = \sqrt{\frac{1}{2}(\underline{\boldsymbol{D}} : \underline{\boldsymbol{D}})} = \sqrt{\frac{1}{2}\operatorname{tr}(\underline{\boldsymbol{D}}\ \underline{\boldsymbol{D}}^T)} = \sqrt{\frac{1}{2}\mathbf{D}_{(II)}} = \frac{1}{\sqrt{2}}\|\underline{\boldsymbol{D}}\|_F \tag{4}$$

where $\|\cdot\|_F$ represents the Frobenius matrix norm.

The power-law model, first proposed by *Ostwald* in 1925, concerns fluids subject to a variation of the shear stress $\dot{\tau}$ proportional (with a proportionality constant K) to a power of the shear rate $\dot{\gamma}$ (see *e.g.* [6]):

$$\dot{\tau} = K\dot{\gamma}^{\alpha},\ \alpha > 0 \tag{5}$$

We thus obtain for the apparent viscosity η, defined as the ratio between shear stress and shear rate:

$$\eta = K\dot{\gamma}^{\alpha-1},\ \alpha > 0 \tag{6}$$

Then, an exponent $\alpha < 1$ corresponds to a *shear-thinning* behavior: apparent viscosity decreases when shear rate increases (*e.g.* blood, lava or ice). On the contrary, an exponent $\alpha > 1$ corresponds to a *shear thickening* behavior: viscosity increases when shear rate increases (*e.g.* corn starch suspensions or certain cement pastes). The case $\alpha = 1$ corresponds to a Newtonian behavior.

Considering a *power-law constitutive equation*, according to (6), writing $s = \alpha+1$, the deviatoric tensor becomes:

$$\underline{\boldsymbol{S}} = \underline{\boldsymbol{\sigma}} + p\underline{\boldsymbol{Id}} = 2\eta_0\|\underline{\boldsymbol{D}}\|_F^{s-2}\underline{\boldsymbol{D}} \tag{7}$$

where $\eta_0 = \dfrac{K}{\sqrt{2}^{s-2}}$ is the apparent viscosity.

We consider in the sequel a domain of computation Ω_t open, bounded and connected in $\mathbb{R}^2$ with Lipschitz-continuous boundary $\partial\Omega$. The conservation of the momentum is described by the Navier-Stokes equation with the low Reynolds number approximation defined on Ω_t:

$$-\boldsymbol{div}(\underline{\boldsymbol{\sigma}}) = f,\ \in \Omega \tag{8}$$

The assumption of *incompressibilty* leads to the following mass-conservation equation:

$$\boldsymbol{div}(\mathbf{u}) = 0,\ \in \Omega \tag{9}$$

where $\mathbf{u}$ represents the velocity field, ρ the density and f an external body force. The following considers bidimensional vertical flow in (x, z) so that $\mathbf{u} = (u_x, u_z)$. It extends naturally, in terms of equations and method, to three dimensions.

For the transient problem, the surface Γ_s, considered to be the graph of a function $h(x, t)$, is a free-surface moving in time. Its movement is described by the following monodimensional transport equation:

$$\partial_t h + u_x\partial_x h = a + u_z \quad \text{on } \Gamma_s \tag{10}$$

with $\mathbf{u} = (u_x(x,z,t), u_z(x,z,t)) \ \ \forall (x,z) \in \Omega_t, \ t \in [0,T]$ and Ω_t is the domain at time t. A mass-balance at surface is represented by the source term a. The present study focuses on a free-surface problem at time given fixed, and thus the evolution of the domain with time is not addressed here.

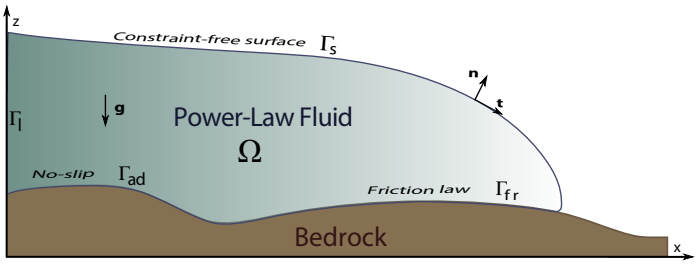


FIGURE 1. *Typical geometry of a 2D vertical free-surface geophysical flow with a front on a general bedrock and notations.*

The boundary $\partial\Omega$ is divided as follows:

- Γ_s is the upper surface (free-surface)
- Γ_b is the bottom surface
- Γ_l and Γ_r are lateral open boundaries.

We introduce $(\boldsymbol{\tau}, \boldsymbol{n})$, the tangent-normal pair of unit vectors such that:

$$\underline{\boldsymbol{\sigma}} = (\underline{\boldsymbol{\sigma}} \cdot \boldsymbol{n})\boldsymbol{n} + (\underline{\boldsymbol{\sigma}} \cdot \boldsymbol{\tau})\boldsymbol{\tau} \tag{11}$$

with:

$$\underline{\boldsymbol{\sigma}} \cdot \boldsymbol{n} = \sigma_{nn}\boldsymbol{n} + \sigma_{n\tau}\boldsymbol{\tau} \quad , \quad \underline{\boldsymbol{\sigma}} \cdot \boldsymbol{\tau} = \sigma_{\tau n}\boldsymbol{n} + \sigma_{\tau\tau}\boldsymbol{\tau} \tag{12}$$

One splits the boundary Γ_b into two parts Γ_{fr} and Γ_{ad} corresponding to an homogeneous Dirichlet area ($u_{\tau|_{\Gamma_{ad}}} = 0$, $u_{n|_{\Gamma_{ad}}} = 0$) and a basal sliding area (see Figure 1). We consider a sliding law (also known as linear Weertman-type sliding law, see [48]) on Γ_{fr}:

$$\begin{cases} \sigma_{\mathbf{n}\tau} = -\beta u_\tau & (13) \\ \boldsymbol{u} \cdot \boldsymbol{n}_{|\Gamma_b} = 0 & (14) \end{cases}$$

where $\beta \geq 0$ is referred to as the friction coefficient.

The upper-surface Γ_s is subject to a stress-free boundary condition: $-\underline{\boldsymbol{\sigma}}\mathbf{n}_{|\Gamma_s} = 0$. The lateral boundary Γ_l is subject to a Neumann boundary condition such as: $-\underline{\boldsymbol{\sigma}}\boldsymbol{n}_{|\Gamma_l} = p_l$. A typical domain is represented on Figure 1.

3. Basic Numerical Procedures. This section presents classical numerical treatments of the quasi-static Stokes problem. We first introduce the variational formulation of problem (8)-(9) and the corresponding mixed finite element discretization. The solution of the linear case and the treatment of the friction condition are then shortly described followed by the two classical methods applied to solve the nonlinear weak Stokes problem (22), that are the fixed point method and the Newton-Raphson method. These basic procedures and formulations as well as the actual drawbacks of the fixed point and the Newton method for the present power-law Stokes problem are required to introduce and elaborate the augmented Lagrangian algorithm ***LA***.

3.1. Mixed Weak Formulation. The variational form of the Stokes problem (8)-(9) with Ω_t given is classical in the Newtonian case and is written as a well-posed mixed variational formulation in well-chosen Hilbert spaces. Hereafter is presented the standard weak formulation of the present power-law Stokes problem (see *e.g.* [7] for the homogeneous Dirichlet case and [25] for more general boundary conditions). Existence and uniqueness results for a given domain Ω based on the minimal dissipation form of the problem are recalled. The steady quasi-static Stokes problem is written:

Find $(\mathbf{u}, p)$ such that

$$\begin{cases} -\boldsymbol{div}(\underline{\boldsymbol{\sigma}}) = f \ \text{ in } \Omega & (15) \\ \boldsymbol{div}(\mathbf{u}) = 0 & (16) \end{cases}$$

with $\underline{\boldsymbol{\sigma}}$ defined by (2).

For notation simplicity, we consider the boundary terms as one integral on a part of the boundary $\Gamma = \Gamma_r \cup \Gamma_l \subset \partial\Omega$ of a unique surface force f_s such that:

$$f_s = \begin{cases} p_l \text{ on } \Gamma_l \\ p_r \text{ on } \Gamma_r \end{cases} \tag{17}$$

Let us consider a test function $\varphi \in X_s^0$ with:

$$X_s = (W^{1,s}(\Omega))^2, \quad X_s^0 = \{\varphi, \varphi \in X_s, \varphi = 0 \text{ on } \Gamma_{ad}\} \tag{18}$$

where s is the exponent from the power law (7) (where $1 < s < 2$ stands for shear-thinning and $s > 2$ for shear-thickening fluid). Then X_s is a Banach space, $X_s^0 \subset X_s$ and if $s = 2$ (Newtonian case), X_2 is the Hilbert space H^1 and X_2^0 is the Hilbert space H^1 with a prescribed Dirichlet boundary condition.

The Green's formula applied to the weak form of the momentum equation (15) yields:

$$\int_\Omega \underline{\boldsymbol{\sigma}} : \underline{\boldsymbol{D}}(\varphi)\ d\boldsymbol{x} - \int_\Gamma f_s \cdot \varphi\ ds + \int_{\Gamma_{fr}} \underline{\boldsymbol{\sigma}}\boldsymbol{n} \cdot \varphi\ ds = \int_\Omega f.\varphi\ d\boldsymbol{x},\ \forall \varphi \in X_s \tag{19}$$

where :

$$\int_{\Gamma_{fr}} \underline{\boldsymbol{\sigma}}\boldsymbol{n} \cdot \varphi\ ds = \int_{\Gamma_{fr}} \sigma_{nn}(\varphi \cdot \boldsymbol{n})\ ds + \int_{\Gamma_{fr}} \sigma_{n\tau}\boldsymbol{\tau} \cdot \varphi\ ds = -\int_{\Gamma_{fr}} \beta u_\tau \cdot \varphi\ ds \tag{20}$$

according to (13) and where $(\underline{\boldsymbol{\sigma}} : \nabla\varphi)$ represents the tensorial scalar product (which is a double contraction) inducing the Frobenius matrix norm: $\underline{\boldsymbol{\sigma}} : \underline{\boldsymbol{\sigma}} = \|\underline{\boldsymbol{\sigma}}\|_F^2 = \langle \boldsymbol{\sigma}, \boldsymbol{\sigma} \rangle_F$, $\underline{\boldsymbol{\sigma}}$ defined by (2). Let us define the pressure space M_{s^*} by:

$$M_{s^*} = L_0^{s^*}(\Omega) = \{q,\ q \in L^{s^*}(\Omega), \int_\Omega q\ d\boldsymbol{x} = 0\} \tag{21}$$

with s^* the conjugate exponent of s *i.e.* $ss^* = s + s^*$. Note that $M_2 = L_0^2$ in the Newtonian case ($s = 2$). We assume in the following that the surface forces f_s belong to $L^{s^*}(\Gamma)$ and the volume force f belongs to $L^{s^*}(\Omega)$.

From the expression of the stress tensor (2), the mixed weak formulation of the Stokes problem (15)-(16) is eventually obtained:

$$\begin{cases} \text{Find } (\mathbf{u}, p) \in X_s^0 \times M_{s^*} \ \text{ such that} \\ a(\eta(\mathbf{u}); \mathbf{u}, \varphi) + b(\varphi, p) = l(\varphi) \ \ \forall \varphi \in X_s \\ b(\mathbf{u}, q) = 0 \ \ \forall q \in M \end{cases} \tag{22}$$

where:

$$a(\eta(\mathbf{u}); \mathbf{u}, \varphi) = \int_\Omega 2\eta(\mathbf{u})\underline{\boldsymbol{D}}(\mathbf{u}) : \underline{\boldsymbol{D}}(\varphi)\ d\boldsymbol{x} - \int_\Gamma f_s \cdot \varphi\ ds - \int_{\Gamma_{fr}} \beta u_\tau \cdot \varphi\ ds, \tag{23}$$

$$b(\mathbf{u}, q) = -\int_\Omega q(\nabla \cdot \mathbf{u})\ d\boldsymbol{x}, \tag{24}$$

$$l(\varphi) = \int_\Omega f.\varphi\ \ d\boldsymbol{x}. \tag{25}$$

From equation (7), we define the *internal viscoplastic dissipation potential* $\mathcal{D}_1$ as follows:

$$\mathcal{D}_1(\underline{\boldsymbol{D}}) = \frac{2\eta_0}{s}\|\underline{\boldsymbol{D}}\|_F^s \tag{26}$$

where s comes from (7) such that:

$$s = \alpha + 1,\ s > 1 \tag{27}$$

and α comes from (6). The $\|.\|_F$ is defined by (4).

The dissipation potential $\mathcal{D}_1$ defined by (26) represents a *Norton-Hoff* viscoplastic model also called *perfect viscoplasticity*; the lack of a threshold stress (also called yield stress) in the law comes to neglect a possible elastic behavior of the material. The Gâteaux differential of $\mathcal{D}_1$ leads the following identity:

$$\mathcal{D}_1' = 2\eta_0\|\underline{\boldsymbol{D}}\|_F^{s-2}\underline{\boldsymbol{D}} = \underline{\boldsymbol{S}} \tag{28}$$

As a consequence, the case of a generic power-law viscous fluid gives for the viscosity η:

$$\eta(\mathbf{u}) = \eta_0\|\underline{\boldsymbol{D}}(\mathbf{u})\|_F^{s-2} \tag{29}$$

If we restrict the space X_s^0 to the space of divergence-free test functions $X_{s,div}^0$:

$$X_{s,div}^0 = \left\{\varphi, \varphi \in X_s^0, \boldsymbol{div}(\varphi) = 0\right\} \tag{30}$$

The dissipation potential $\mathcal{D}_1$ being convex with respect to $\underline{\boldsymbol{D}}$, the problem (22) is formally equivalent to

$$(P)\begin{cases} \text{Find } \mathbf{u} \in X_{s,div}^0 \text{ such that:} \\ J(\mathbf{u}) = \min\limits_{\varphi \in X_{s,div}^0} J(\varphi) \end{cases} \tag{31}$$

where $J(.)$ is the functional defined by:

$$J(\varphi) = \int_\Omega \mathcal{D}_1(\underline{\boldsymbol{D}}(\varphi))\ d\boldsymbol{x} - \int_\Omega f\cdot\varphi\ d\boldsymbol{x} - \int_\Gamma f_s.\varphi\ ds - \int_{\Gamma_{fr}} \beta u_\tau\cdot\varphi\ ds \tag{32}$$

with $\mathcal{D}_1$ defined by (26).

From a physical point of view, the formulation (31) of the incompressible viscoplastic quasi-static flow model (15)-(16) corresponds to a standard property of the Stokes equation that is the *minimal dissipation*. Thus, solving problem (P) defined by (31) is tantamount to minimizing the viscoplastic rate of energy dissipation $J(\varphi)$ on the space $X_{s,div}^0$ of kinematically admissible velocity fields.

The linear stationary case. The linear stationary problem is described using the classical saddle-point formulation of the Stokes problem (see *e.g.* [43]). Its discrete counterpart is obtained using the standard *Taylor-Hood* $(P_2 - P_1)$ finite element. The non-penetration condition (14) associated to the friction boundary condition is imposed using a Lagrange multiplier.

There are several ways to solve the resulting linear system. A classical method is to use an *Uzawa algorithm* relaxing the divergence-free constraint and the non-penetration constraint. This type of algorithm is robust and converges well. Another approach consists in obtaining a direct solution of the complete matrix problem using for example the numerical library *MUMPS* (see [3]) through its multifrontal linear system solver. Our numerical experiments showed great efficiency with this last approach.

3.2. Classical Solvers and Limitations.

Fixed point method. A classical way to solve the problem (22) is to use an iterative Picard algorithm based on a fixed point method.

At each iteration, the assembly of the Stokes linear system has to be performed and solved using, for example, an Uzawa-type algorithm or a direct solver such as *e.g.* the numerical library *MUMPS* [3]. The Picard scheme provides a rate of convergence at best linear, making it a rather slowly convergent algorithm.

Newton-Raphson method. For the sake of clarity, we skip the boundary terms in the mixed weak problem (23)-(24). One sets:

$$R_1(\mathbf{u},p;\varphi) \quad = \quad 2\eta_0 \int_\Omega (\|\underline{\boldsymbol{D}}(\mathbf{u})\|_F^{s-2}\underline{\boldsymbol{D}}(\mathbf{u})) : \nabla\varphi \; d\boldsymbol{x} + \int_\Omega \nabla p \cdot \varphi \; d\boldsymbol{x} - \int_\Omega f.\varphi \; d\boldsymbol{x} \tag{33}$$

$$R_2(\mathbf{u},p;q) = \int_\Omega q \, \boldsymbol{div}(\mathbf{u}) \; d\boldsymbol{x} \tag{34}$$

The Newton algorithm then writes:

1. Given an initial value (u_0, p_0)
2. For $k \geq 0$, determine a correction $(\delta u_{k+1}, \delta p_{k+1})$ by solving:

$$\left\{ \begin{array}{l} \dfrac{\partial R_1}{\partial \mathbf{u}}(u_k,p_k;\varphi)\cdot\delta u_{k+1} + \dfrac{\partial R_1}{\partial p}(u_k,p_k;\varphi)\cdot\delta p_{k+1} = -R_1(u_k,p_k;\varphi) \\ \dfrac{\partial R_2}{\partial \mathbf{u}}(u_k,p_k;\varphi)\cdot\delta u_{k+1} + \dfrac{\partial R_2}{\partial p}(u_k,p_k;\varphi)\cdot\delta p_{k+1} = -R_2(u_k,p_k;\varphi) \end{array} \right. \tag{35}$$

3. Set $u_{k+1} = u_k - \delta u_{k+1}$ and $p_{k+1} = p_k - \delta p_{k+1}$

where:

$$\begin{array}{c} \dfrac{\partial R_1}{\partial \mathbf{u}}(\mathbf{u}_0,p_0;\varphi)\cdot\delta\mathbf{u} = 2\eta_0 \displaystyle\int_\Omega \|\underline{\boldsymbol{D}}(\mathbf{u}_0)\|_F^{s-2} \left\langle \underline{\boldsymbol{D}}(\delta\mathbf{u}), \nabla\varphi \right\rangle_F \; d\boldsymbol{x} \\ +2\eta_0(s-2)\displaystyle\int_\Omega \|\underline{\boldsymbol{D}}(\mathbf{u}_0)\|_F^{s-4} \left\langle \underline{\boldsymbol{D}}(\mathbf{u}_0), \underline{\boldsymbol{D}}(\delta\mathbf{u}) \right\rangle_F \left\langle \underline{\boldsymbol{D}}(\mathbf{u}_0), \nabla\varphi \right\rangle_F \; d\boldsymbol{x} \end{array} \tag{36}$$

According to equation (36), in the shear-thinning case ($1 < s < 2$), the Newton method is ill-posed since the bilinear form is not defined for any constant $\mathbf{u}_0$ (due to the vanishing of $\|\underline{\boldsymbol{D}}(\mathbf{u}_0)\|$). Then it cannot be theoretically applied (see [26] for similar observations for the solution of the p-Laplacian problem). Beyond this observation, numerical experiments demonstrate a certain ability for the Newton algorithm to reach a converge solution for academic test case (see Section 5). However, the radius of its convergence disk is unknown and we observe numerically that the Newton algorithm generally requires a good first-guess obtained from another method such as the fixed point. The required accuracy for the first-guess is unclear and depends, among others, on the value of the power-law index s. Indeed, in the one-dimensional case, the bilinear form $\dfrac{\partial R_1}{\partial u}$ is such that:

$$\frac{\partial R_1}{\partial u}(\varphi,\varphi) \quad = \quad 2\eta_0(s-1)|u_0'|^{s-2}\|\varphi'\|_{L^2} \tag{37}$$

Therefore, the bilinear form attached to the Newton-Raphson algorithm degenerates when s tends to 1, which corresponds to an increase of the shear-thinning behavior of the fluid. Numerical estimations of this limitation for strongly pseudoplastic flows are shown in Section 5.

Moreover, the use of these methods (Picard and Newton) to compute an adjoint state based on algorithmic differentiation raises a strong memory issue. Indeed, the evaluation of the adjoint state resulting from an iteratively computed forward state (typically when solving a nonlinear problem) using source code differentiation is obtained by *reverse accumulation* (see [13]). In terms of memory, this procedure leads to store each state the forward iterative solver goes through. It consequently

leads to an important memory burden which can be unacceptable for large scale systems (in particular for parameter identification, see [39] for a detailed description of this memory aspect). We seek hereafter a fast and robust algorithm with small memory consumption in order to solve the power-law Stokes problem in the perspective of extensive finite element adjoint computations.

4. Four-Field Saddle Point Problem. Following [25], a splitting technique on the *minimal dissipation form* (31) is applied to the corresponding variational problem. The splitting consists in introducing the strain-rate tensor $\underline{\boldsymbol{D}}$ as an unknown. It consequently allows to solve the linear part and the nonlinear part of the governing equations sequentially.

This technique applied to the present problem leads to a *four-field saddle point* formulation of the Stokes problem for which we demonstrate the well-posedness. This saddle-point problem is then discretized using a three-field $P_2 - P_1 - P_1^{disc}$ finite element description. We then build an *augmented Lagrangian type algorithm* in order to solve the four-field saddle-point problem.

The resulting algorithm demonstrates its ability to significantly lessen the computational time compared to the fixed point and the Newton methods. At the same time, the P_1 discontinuous discretization of the strain-rate tensor $\underline{\boldsymbol{D}}$ considered here leads to solve a diagonal-block linear system at element level. This convenient structure allows to drastically reduce memory needs and is thus particularly interesting in an adjoint-based data assimilation perspective.

Let us point out that in the more general case of viscoplastic behaviors like the Herschel-Bulkley law that include a yield stress, this type of method is suitable for the corresponding non differentiability appearing in the energy functional (see *e.g.* [33]). Then, the following method can be extended to elastic-viscoplastic fluid.

4.1. Four-Field Saddle-Point Formulation. Let us recall that the model (15)-(16) is formally equivalent to the minimization problem (P) defined by (31) which admits a unique solution $(\mathbf{u}, p)$ in $(X^0_{s,div} \times M_{s^*})$ (respectively defined by (18) and (21)). In addition, the existence of the deviatoric stress tensor field $\underline{\boldsymbol{S}}$ is obtained in the space H_{s^*} defined by:

$$H_{s^*} = [L^{s^*}(\Omega)]^{2\times 2} \tag{38}$$

Thus, we set:

$$V = \left\{(\mathbf{u}, \underline{\boldsymbol{d}}) \in [W^{1,s}(\Omega)]^2 \times [L^s(\Omega)]^{2\times 2}\right\} \tag{39}$$

$$Z = \left\{(p, \underline{\boldsymbol{\tau}}) \in L^{s^*}(\Omega) \times [L^{s^*}(\Omega)]^{2\times 2}\right\} \tag{40}$$

and

$$\mathcal{A}_s = (X^0_s \times H_s) \subset V \tag{41}$$

$$\mathcal{B}_{s^*} = (M_{s^*} \times H_{s^*}) \subset Z \tag{42}$$

We denote hereafter by $< .,. >$, the duality product in the corresponding spaces X^0_s, M_{s^*}, H_s or H_{s^*}.

Then, we decompose the functional J defined by (32) as follows:

$$J(\mathbf{u}) = F(\underline{\boldsymbol{D}}(\mathbf{u})) + G(\mathbf{u}) \tag{43}$$

with $F : H_s \to \overline{\mathbb{R}}$, $G : X^0_s \to \overline{\mathbb{R}}$ convex, proper (not identically infinites), continuous functions defined by:

$$F(\underline{\boldsymbol{d}}) = \frac{2\eta_0}{s} \int_\Omega \|\underline{\boldsymbol{d}}\|^s_F \, d\boldsymbol{x} \tag{44}$$

$$G(\mathbf{u}) = -\int_\Omega f \cdot \mathbf{u}\, d\boldsymbol{x} \tag{45}$$

Using these definitions, the decomposition (43) holds. We then define $W \subset \mathcal{A}_s$ by:

$$W = \{(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s, \boldsymbol{div}(\mathbf{u}) = 0, \underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{d}} = 0\} \tag{46}$$

with $\underline{\boldsymbol{D}}$ defined by (3). Then, problem (P) defined by (31) is formally equivalent to:

$$(\Pi) \left\{ \begin{array}{l} \text{Find } (\mathbf{u}^*, \underline{\boldsymbol{d}}^*) \in W \text{ such that:} \\ j(\mathbf{u}^*, \underline{\boldsymbol{d}}^*) = \min\limits_{(\mathbf{u}, \underline{\boldsymbol{d}}) \in W} j(\mathbf{u}, \underline{\boldsymbol{d}}) \end{array} \right. \tag{47}$$

where:

$$j(\mathbf{u}, \underline{\boldsymbol{d}}) = F(\underline{\boldsymbol{d}}) + G(\mathbf{u}) \tag{48}$$

Problem (Π) introduces a new variable d and a new constraint:

$$\underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{d}} = 0 \tag{49}$$

The decomposition-coordination method developed in [25] as applied to the present fluid model, is based on the *equivalency* between Problem (P) and Problem (Π). For the sake of simplicity, we consider in the sequel only zero boundary terms, that is to say: $f_s = 0$, f_s defined by (17) and $\beta = 0$, β defined by (13). Nevertheless, the extension to general boundary condition simulations is natural and numerical computations with Neumann and friction boundary conditions are presented in section 5.3.

Saddle-point formulation of problem (Π). Let us introduce the functional $\mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}}))$ defined on $(\mathcal{A}_s \times \mathcal{B}_{s^*})$ by:

$$\mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) = j(\mathbf{u}, \underline{\boldsymbol{d}}) - < p, \boldsymbol{div}(\mathbf{u}) > + < \underline{\boldsymbol{\tau}}, \underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{d}} > \tag{50}$$

The functional $\mathcal{L}$ corresponds to the Lagrangian of the problem (Π) defined by (47) with respect to the two equality constraints (16) and (49).

4.2. Existence and Uniqueness of the Saddle-Point. The duality framework of the saddle-point formulation is presented below. On the one hand, using a duality theorem, we ensure existence of a saddle-point for the four-field Lagrangian defined by (50) in the Banach space $\mathcal{A}_s \times \mathcal{B}_{s^*}$. On the other hand, from the existence of the saddle-point and the existence and uniqueness of the solution of the minimization problem (P), the uniqueness of the pair $(\mathbf{u}^*, \underline{\boldsymbol{d}}^*)$ of the saddle-point of the Lagrangian (50) is ensured but not the uniqueness of the Lagrange multipliers part $(p^*, \underline{\boldsymbol{\tau}}^*)$.

Let us consider $\mathcal{L} : (\mathcal{A}_s \times \mathcal{B}_{s^*}) \to \mathbb{R}$ defined by (50). For $(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s$ and $(p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*}$, one sets:

$$\mathcal{J}(\mathbf{u}, \underline{\boldsymbol{d}}) = \sup_{(p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*}} \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) \tag{51}$$

$$\mathcal{G}(p, \underline{\boldsymbol{\tau}}) = \inf_{(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s} \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) \tag{52}$$

One defines then the *primal problem*:

$$\inf_{(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s} \mathcal{J}(\mathbf{u}, \underline{\boldsymbol{d}}) \tag{53}$$

and the *dual problem*:

$$\sup_{(p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*}} \mathcal{G}(p, \underline{\boldsymbol{\tau}}) \tag{54}$$

In this context, a *duality theorem* is recalled:

THEOREM 4.1. *(see [16] p.164) Let V and Z be two reflexive Banach spaces. We assume that:*

$$\mathcal{A}_s \subset V \text{ is convex, closed, non empty} \tag{55}$$
$$\mathcal{B}_{s^*} \subset Z \text{ is convex, closed, non empty} \tag{56}$$

We assume as well that the functional $\mathcal{L} : \mathcal{A}_s \times \mathcal{B}_{s^*} \to \mathbb{R}$ satisfies:

$$\begin{array}{l}\forall (\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s, (p, \underline{\boldsymbol{\tau}}) \mapsto \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) \\ \text{is concave upper semi-continuous}\end{array} \tag{57}$$

$$\begin{array}{l}\forall (p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*}, (\mathbf{u}, \underline{\boldsymbol{d}}) \mapsto \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) \\ \text{is convex lower semi-continuous}\end{array} \tag{58}$$

We also requires that:

$$\begin{array}{l}\exists (p_0, \tau_0) \in \mathcal{B}_{s^*} \text{ such that :} \\ \lim\limits_{\substack{(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s \\ \|(\mathbf{u}, \underline{\boldsymbol{d}})\| \to \infty}} \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p_0, \underline{\boldsymbol{\tau_0}})) = +\infty\end{array} \tag{59}$$

and

$$\lim_{\substack{(p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*} \\ \|(p, \underline{\boldsymbol{\tau}})\| \to \infty}} \inf_{(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s} \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) = -\infty \tag{60}$$

Then, $\mathcal{L}$ admits at least one saddle-point $((\mathbf{u}^*, \underline{\boldsymbol{d}}^*), (p^*, \underline{\boldsymbol{\tau}}^*))$ on $\mathcal{A}_s \times \mathcal{B}_{s^*}$ which satisfies:

$$\begin{array}{rcl}\mathcal{L}((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*)) & = & \min\limits_{(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s} \sup\limits_{(p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*}} \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) \\ & = & \max\limits_{(p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*}} \inf\limits_{(\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s} \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}}))\end{array} \tag{61}$$

PROPOSITION 4.2. *With $\mathcal{A}_s$ and $\mathcal{B}_{s^*}$ respectively defined by* (41) *and* (42) *and $\mathcal{L}$ defined by* (50)*, there exists a saddle-point $((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*)) \in \mathcal{A}_s \times \mathcal{B}_{s^*}$ which satisfies* (61)*.*

Proof. Spaces L^p and $W^{1,p}$ are uniformly convex for $1 < p < \infty$. A cartesian product of uniformly convex Banach spaces being uniformly convex, $\mathcal{A}_s$ and $\mathcal{B}_{s^*}$ are uniformly convex (see *e.g* [1]). The space $\mathcal{A}_s$ (resp. $\mathcal{B}_{s^*}$) being a complete subspace of a metric space V (resp. Z), $\mathcal{A}_s$ (resp. $\mathcal{B}_{s^*}$) is *closed*. Then hypothesis (55) and (56) holds.

Besides, $\mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}}))$ defined by (50) is convex with respect to $(\mathbf{u}, \underline{\boldsymbol{d}})$ from the strict convexity of the function $F(\underline{\boldsymbol{d}})$ and the linearity of the other terms, and linear with respect to $(p, \underline{\boldsymbol{\tau}})$ and then concave. Moreover, $\mathcal{L}$ is Gâteaux-differentiable with respect to each variable and subsequently *continuous*, hence satisfying hypothesis (57) and (58).

We recall the complete expression of the Lagrangian:

$$\mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) = \frac{2\eta_0}{s} \int_\Omega \|\underline{\boldsymbol{d}}\|_F^s \, d\boldsymbol{x} - \int_\Omega f \cdot \mathbf{u} \, d\boldsymbol{x} - < p, \boldsymbol{div}(\mathbf{u}) > + < \tau, \underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{d}} > \tag{62}$$

Proof of hypothesis (59). One chooses $(p_0, \underline{\boldsymbol{\tau_0}})$ such that:

$$-\boldsymbol{div}(\underline{\boldsymbol{\tau_0}}) + \nabla p_0 = f \tag{63}$$

The weak form of (63) becomes:

$$\int_\Omega -\boldsymbol{div}(\underline{\boldsymbol{\tau_0}}) \cdot \varphi \, d\boldsymbol{x} + \int_\Omega \nabla p_0 \cdot \varphi \, d\boldsymbol{x} = \int_\Omega f \cdot \varphi, \ \forall \varphi \in X_s^0 \tag{64}$$

Using integration by part this expression becomes:

$$\int_\Omega \underline{\boldsymbol{\tau_0}} : \underline{\boldsymbol{D}}(\varphi)\ d\boldsymbol{x} - \int_\Omega p_0\, \boldsymbol{div}(\varphi)\ d\boldsymbol{x} - \int_\Omega f\cdot\varphi = 0,\ \forall \varphi \in X_s^0 \tag{65}$$

Since $\mathbf{u} \in X_s^0$, we have:

$$\mathcal{L}((\mathbf{u},\underline{\boldsymbol{d}});(p_0,\underline{\boldsymbol{\tau_0}})) = \frac{2\eta_0}{s}\int_\Omega \|\underline{\boldsymbol{d}}\|_F^s\ d\boldsymbol{x} - <\underline{\boldsymbol{\tau_0}}, d> \tag{66}$$

The proper, continuous and convex ($s>1$) functional $f(\underline{\boldsymbol{d}})$ is defined by :

$$f(\underline{\boldsymbol{d}}) = \int_\Omega \|\underline{\boldsymbol{d}}\|_F^s\ d\boldsymbol{x} = \|\underline{\boldsymbol{d}}\|_{H_s}^s \tag{67}$$

We thus obtain:

$$\lim_{\|\underline{\boldsymbol{d}}\|_{H_s}\to\infty} \frac{f(\underline{\boldsymbol{d}})}{\|\underline{\boldsymbol{d}}\|_{H_s}} = \lim_{\|\underline{\boldsymbol{d}}\|_{H_s}\to\infty} \|\underline{\boldsymbol{d}}\|_{H_s}^{s-1} = +\infty \tag{68}$$

hence the *strong coercivity* (or *supercoercivity*) on f.

We also consider the proper, continuous and linear functional $g(\underline{\boldsymbol{d}}) : H_s \to \bar{\mathbb{R}}$ defined by:

$$g(\underline{\boldsymbol{d}}) = -<\underline{\boldsymbol{\tau_0}}, \underline{\boldsymbol{d}}> \tag{69}$$

From the continuity of g, one can deduce:

$$g(\underline{\boldsymbol{d}}) \geq -C\|\underline{\boldsymbol{d}}\|_{H_s} \tag{70}$$

Then,

$$\frac{f(\underline{\boldsymbol{d}}) + g(\underline{\boldsymbol{d}})}{\|\underline{\boldsymbol{d}}\|_{H_s}} \geq \|\underline{\boldsymbol{d}}\|_{H_s}^{s-1} - C \underset{\|\underline{\boldsymbol{d}}\|_{H_s}\to\infty}{\longrightarrow} +\infty \tag{71}$$

from what we deduce the supercoercivity of $\mathcal{L}((\mathbf{u},\underline{\boldsymbol{d}});(p_0,\underline{\boldsymbol{\tau_0}}))$:

$$\lim_{\substack{(\mathbf{u},\underline{\boldsymbol{d}})\in\mathcal{A}_s \\ \|(\mathbf{u},\underline{\boldsymbol{d}})\|\to\infty}} \mathcal{L}((\mathbf{u},\underline{\boldsymbol{d}});(p_0,\underline{\boldsymbol{\tau_0}})) = +\infty \tag{72}$$

Therefore, hypothesis (59) holds.

Proof of hypothesis (60). Let us consider the pair $(\mathbf{u}_0,\underline{\boldsymbol{d}}_0)$ such that $\boldsymbol{div}(\mathbf{u}_0) = 0$ and $\underline{\boldsymbol{D}}(\mathbf{u}_0) = \underline{\boldsymbol{d}}_0$. One gets:

$$\mathcal{L}((\mathbf{u}_0,\underline{\boldsymbol{d}}_0);(p,\underline{\boldsymbol{\tau}})) = \frac{2\eta_0}{s}\int_\Omega \|\underline{\boldsymbol{d}}_0\|_F^s\ d\boldsymbol{x} - \int_\Omega f\cdot\mathbf{u}_0\ d\boldsymbol{x},\ \forall (p,\underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*} \tag{73}$$

from what we deduce:

$$\inf_{(\mathbf{u},\underline{\boldsymbol{d}})\in\mathcal{A}_s} \mathcal{L}((\mathbf{u},\underline{\boldsymbol{d}});(p,\underline{\boldsymbol{\tau}})) = -\infty \tag{74}$$

Then, hypothesis (60) holds.

Finally, $\mathcal{L}$ defined by (50) satisfies every hypothesis of Theorem 4.1 and the duality theorem holds. □

About the uniqueness of the saddle-point. The functional $\mathcal{J}(\mathbf{u}, \underline{\boldsymbol{d}})$ minimized in the primal problem (53) corresponds, under equality constraints (16) and (49), to the functional $j(\mathbf{u}, \underline{\boldsymbol{d}})$ defined by (48). The primal problem (53) related to the Lagrangian then corresponds to the minimization problem (Π) defined by (47) which admits a unique minimum $(\mathbf{u}^*, \underline{\boldsymbol{d}}^*)$.

Moreover, according to (61), $\mathcal{G}(p, \underline{\boldsymbol{\tau}}) \leq \mathcal{J}(\mathbf{u}, \underline{\boldsymbol{d}}),\ \forall (\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A}_s$. We deduce that:

$$\mathcal{G}(p, \underline{\boldsymbol{\tau}}) \leq \mathcal{J}(\mathbf{u}^*, \underline{\boldsymbol{d}}^*) = \mathcal{G}(p^*, \underline{\boldsymbol{\tau}}^*),\ \forall (p, \underline{\boldsymbol{\tau}}) \in \mathcal{B}_{s^*} \tag{75}$$

It amounts to saying that $(p^*, \underline{\boldsymbol{\tau}}^*)$ is a maximum of $\mathcal{G}$. It implies that the domain of the dual problem (54) is not empty; nevertheless, the dual problem does not have a unique solution *a priori* due to the linearity of the functional $(p, \underline{\boldsymbol{\tau}}) \mapsto \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}}))$. Thus $\mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}}))$ does not have *a priori* a unique saddle-point.

The functional spaces used for the previous continuous development depend on the rheological exponent s. Consequently, if $s \neq 2$, these functional spaces are not Hilbert spaces. In principle, if $s \neq 2$, the augmented Lagrangian approach discussed in [24] does not apply to these non-Hilbertian situations. It applies however to those finite dimensional spaces obtained by the finite element approximation of the continuous problem. In what follows, we set $X^0 = X_2^0$, $M = M_2$ and $H = H_2$ and consequently $\mathcal{A} = \mathcal{A}_2$ and $\mathcal{B} = \mathcal{B}_2$ (defined respectively by (18), (21), (38), (39) and (42)).

4.3. Characterization of the saddle-point. The saddle-point existence result allows to characterize the solution as follows:

PROPOSITION 4.3. *(see e.g.[16]) Assuming hypothesis* (55)-(58) *and using the Gâteaux differentiability of* $\mathcal{L}$ *with respect to* $(\mathbf{u}, \underline{\boldsymbol{d}})$ *and* $(p, \underline{\boldsymbol{\tau}})$*, the existence of the saddle-point* $((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*))$ *is equivalent to:*

$$\left\langle \frac{\partial \mathcal{L}}{\partial (\mathbf{u}, \underline{\boldsymbol{d}})}((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*)), (\mathbf{u}, \underline{\boldsymbol{d}}) - (\mathbf{u}^*, \underline{\boldsymbol{d}}^*) \right\rangle \geq 0,\ \forall (\mathbf{u}, \underline{\boldsymbol{d}}) \in \mathcal{A} \tag{76}$$

$$\left\langle \frac{\partial \mathcal{L}}{\partial (p, \underline{\boldsymbol{\tau}})}((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*)), (p, \underline{\boldsymbol{\tau}}) - (p^*, \underline{\boldsymbol{\tau}}^*) \right\rangle \leq 0,\ \forall (p, \underline{\boldsymbol{\tau}}) \in \mathcal{B} \tag{77}$$

This proposition characterizes the saddle-point using *elliptic variational inequalities (EVI)* of second kind. Using the Gâteaux differentiability of the Lagrangian, these EVIs corresponds, in fact, to vanishing Lagrangian gradients with respect to the unknows and the Lagrange multipliers. The existence of the saddle-point $((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*))$ is then equivalent to:

$$\left\{ \begin{array}{l} \dfrac{\partial \mathcal{L}}{\partial (\mathbf{u}, \underline{\boldsymbol{d}})}((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*)) = 0 \\ \dfrac{\partial \mathcal{L}}{\partial (p, \underline{\boldsymbol{\tau}})}((\mathbf{u}^*, \underline{\boldsymbol{d}}^*); (p^*, \underline{\boldsymbol{\tau}}^*)) = 0 \end{array} \right. \tag{78}$$

where the derivative with respect to the unknowns represents the necessary optimality condition and the derivative with respect to the Lagrange multiplier represents the constraints.

The second equation of system (78) yields:

$$\frac{\partial \mathcal{L}}{\partial p} \cdot \delta p = - < \boldsymbol{div}(\mathbf{u}), \delta p >= 0\ ,\ \forall \delta p \in M \tag{79}$$

$$\frac{\partial \mathcal{L}}{\partial \underline{\boldsymbol{\tau}}} \cdot \delta \underline{\boldsymbol{\tau}} =< \underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{d}}, \delta \underline{\boldsymbol{\tau}} >= 0\ ,\ \forall \delta \underline{\boldsymbol{\tau}} \in H \tag{80}$$

It corresponds to the two constraints (16) and (49) of Problem (Π) defined by (47). The first equation of system (78) yields:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} \cdot \delta\mathbf{u} = -\int_\Omega f \cdot \delta\mathbf{u}\ d\boldsymbol{x} - < p, \boldsymbol{div}(\delta\mathbf{u}) > + < \underline{\boldsymbol{\tau}}, \underline{\boldsymbol{D}}(\delta\mathbf{u}) >= 0\ ,\ \forall \delta\mathbf{u} \in X^0 \tag{81}$$

$$\frac{\partial \mathcal{L}}{\partial \underline{\boldsymbol{d}}} \cdot \delta\underline{\boldsymbol{d}} = 2\eta_0 \int_\Omega (\|\underline{\boldsymbol{d}}\|_H^{s-2} < \underline{\boldsymbol{d}}, \delta\underline{\boldsymbol{d}} >)\ d\boldsymbol{x} - < \underline{\boldsymbol{\tau}}, \delta\underline{\boldsymbol{d}} >= 0\ ,\ \forall \delta\underline{\boldsymbol{d}} \in H \tag{82}$$

The last two equations can be written as the following coupled system:

$$\left\{ \begin{array}{r} < \nabla p, \delta\mathbf{u} > + < \boldsymbol{D}^* \underline{\boldsymbol{\tau}}, \delta\mathbf{u} > = < f, \delta\mathbf{u} >,\ \forall \delta\mathbf{u} \in X^0 \\ < 2\eta_0 \|\underline{\boldsymbol{d}}\|_H^{s-2} \underline{\boldsymbol{d}}, \delta\underline{\boldsymbol{d}} > = < \underline{\boldsymbol{\tau}}, \delta\underline{\boldsymbol{d}} >,\ \forall \delta\underline{\boldsymbol{d}} \in H \end{array} \right. \tag{83}$$

where $\boldsymbol{D}^*$ represents the adjoint operator of the rate of deformation tensor $\underline{\boldsymbol{D}}$:

$$< \boldsymbol{D}^* \xi, \varphi >= \int_\Omega \xi : \underline{\boldsymbol{D}}(\varphi)\ \ d\boldsymbol{x},\ \xi \in H \tag{84}$$

The second equation of the system (83) corresponds to the weak form of the rheological law (7):

$$< 2\eta_0 \|\underline{\boldsymbol{d}}\|_H^{s-2} \underline{\boldsymbol{d}}, \delta\underline{\boldsymbol{d}} >=< \underline{\boldsymbol{\tau}}, \delta\underline{\boldsymbol{d}} >, \forall \delta\underline{\boldsymbol{d}} \in H \tag{85}$$

LEMMA 4.4. *In the case of a zero boundary term (i.e. $f_s = 0$, f_s defined by (17) and $\beta = 0$, β defined by (13)), the adjoint operator $\boldsymbol{D}^*$ of the rate of deformation tensor $\underline{\boldsymbol{D}}$ applied to a tensor $\boldsymbol{T}$ is the opposite of the $\boldsymbol{div}$ operator applied to the symmetrical part of $\boldsymbol{T}$:*

$$\boldsymbol{D}^*(\boldsymbol{T}) = -\boldsymbol{div}\left(\frac{1}{2}(\boldsymbol{T} + \boldsymbol{T}^T)\right) \tag{86}$$

Proof. : this property is easily deduced from the classical property of the adjoint operator of the gradient operator. □

REMARK 1. *The image of the symmetric tensor $\underline{\boldsymbol{S}}$ by $\boldsymbol{D}^*$ is:*

$$\boldsymbol{D}^*(\underline{\boldsymbol{S}}) = -\boldsymbol{div}(\underline{\boldsymbol{S}}) \tag{87}$$

in the case of a zero boundary term. This result is actually sufficient for any boundary term if one considers a discontinuous discretization for the tensorial unknowns such as the one considered in Section 4.5; indeed discontinuous degrees of freedom do not appear on the boundaries and no boundary condition applies on the resulting linear system.

PROPOSITION 4.5. *The tensorial Lagrange multiplier $\underline{\boldsymbol{\tau}}$ corresponds to the extra-stress tensor $\underline{\boldsymbol{S}}$ and the scalar Lagrange multiplier p to the pressure field.*

Proof. : From Lemma 4.4, we rewrite the first equation of the system (83) as follows:

$$< -\boldsymbol{div}(\frac{1}{2}(\underline{\boldsymbol{\tau}} + \underline{\boldsymbol{\tau}}^T)), \delta\mathbf{u} > + < \nabla p, \delta\mathbf{u} >=< f, \delta\mathbf{u} >, \forall \delta\mathbf{u} \in X^0 \tag{88}$$

And yet, there exists a unique field $(\mathbf{u}, p) \in (X^0_{div} \times H)$ solution of the momentum equation of the Stokes problem such that

$$-\boldsymbol{div}(\underline{\boldsymbol{\sigma}}(\mathbf{u}, p)) = -\boldsymbol{div}(\underline{\boldsymbol{S}}(\mathbf{u})) + \nabla p = f \tag{89}$$

Thus, according to (89), the tensor $\underline{\boldsymbol{\tau}}$ represents the extra-stress tensor $\underline{\boldsymbol{S}}(\mathbf{u})$ (which is symmetric):

$$\underline{\boldsymbol{\tau}} = \underline{\boldsymbol{S}}(\mathbf{u}) \tag{90}$$

and the scalar field p corresponds to the Stokes pressure field. □

We develop hereafter an augmented Lagrangian algorithmic approach in order to solve the well-posed four-field saddle point problem (61) derived in this section.

4.4. *LA* and *LAθ* algorithms. Two augmented Lagrangian algorithms, called ***LA*** and ***LAθ*** well-suited for the four-field saddle-point problem (61) are built. This type of algorithm is a regularized version of the Uzawa algorithm which is a duality-based algorithm that uses the good properties of the dual problem (54) (see *e.g.* [2]). A quadratic regularization term (linked to the Moreau-Yosida regularization) is added to the Lagrangian $\mathcal{L}$ with respect to the constraints of the problem, bringing a new functional called *augmented Lagrangian*, denoted $\mathcal{L}_r$. This functional has the same saddle-point as the initial Lagrangian, for all $r \geq 0$, and its associated dual function $\mathcal{G}_r(p, \underline{\boldsymbol{\tau}})$ has a better conditionning.

The additional terms in the Lagrangian $\mathcal{L}$ defined by (50) yields the following *augmented Lagrangian* $\mathcal{L}_r$:

$$\mathcal{L}_r((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) = \mathcal{L}((\mathbf{u}, \underline{\boldsymbol{d}}); (p, \underline{\boldsymbol{\tau}})) + \frac{r}{2} \int_\Omega |\underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{d}}|^2 d\boldsymbol{x} + \frac{r}{2} \int_\Omega |\, \boldsymbol{div}(\mathbf{u})|^2 \, d\boldsymbol{x} \tag{91}$$

Then, an Uzawa-type algorithm is built with respect to $\mathcal{L}_r$ providing ***LA*** algorithm:

LA: *Let (d^0, p^1, τ^1) be given. For $n \geq 1$, solve:*

$$\begin{cases} -r\boldsymbol{D}^*\underline{\boldsymbol{D}}(u^n) - \nabla p^n = -f + \boldsymbol{D}^*(\tau^n) - r\boldsymbol{D}^*(d^{n-1}) \\ \boldsymbol{div}(u^n) = 0 \end{cases} \tag{92}$$

For $|d^n|_0$ given, $\boldsymbol{k} \geq 0$, compute:

$$|d^n|_{\boldsymbol{k+1}} = |d^n|_{\boldsymbol{k}} - \frac{2\eta_0 |d^n|_{\boldsymbol{k}}^{s-1} + r|d^n|_{\boldsymbol{k}} - |r\underline{\boldsymbol{D}}(u^n) + \tau^n|}{(s-1)2\eta_0 |d^n|_{\boldsymbol{k}}^{s-2} + r} \tag{93}$$

For $|d^n|$ be given, solve:

$$2\eta_0 |d^n|^{s-2} d^n + r d^n = r\underline{\boldsymbol{D}}(u^n) + \tau^n \tag{94}$$

Update the Lagrange multiplier:

$$\tau^{n+1} = \tau^n + \rho(\underline{\boldsymbol{D}}(u^n) - d^n) \tag{95}$$

This algorithm slightly differs from a standard Uzawa since it uses a block-relaxation method which consists in solving the system in two steps by replacing the unknown d^n by d^{n-1} in the first equation of system (92). In addition, the solution of the nonlinear equation (94) is achieved by taking the norm on both side of the equation (see [25]) to obtain

$$\left|2\eta_0 |d^n|^{s-2} + r\right| |d^n| = |r\underline{\boldsymbol{D}}(u^n) + \tau^n| \tag{96}$$

which is, for $r > 0$ and $2\eta_0 |d^n|^{s-2} > 0$, equal to:

$$2\eta_0 |d^n|^{s-1} + r|d^n| = |r\underline{\boldsymbol{D}}(u^n) + \tau^n| \tag{97}$$

This equation is a one variable nonlinear equation that is solved by a Newton algorithm with the step (93). Once this equation is solved, the value of $|d^n|$ is known and the nonlinear equation becomes a standard linear system .

It is important to precise that this algorithm considers a splitting that allows to satisfy the two constraints (*i.e.* $\boldsymbol{div}(\mathbf{u}) = 0$ and $\underline{\boldsymbol{d}} = \underline{\boldsymbol{D}}(\mathbf{u})$) sequentially. The constraint $\underline{\boldsymbol{d}} = \underline{\boldsymbol{D}}(\mathbf{u})$ is satisfied by the augmented Lagrangian algorithm throughout the nonlinear loop from equation (95) whereas the incompressiblity condition is satisfied at each iteration by solving the system (92), typically using an inner augmented Lagrangian algorithm. The fully decoupled scheme does not work well in practice due to a competition between the satisfaction of the two constraints.

Our numerical experiments show that the best performance are obtained using $\rho = r$, ρ coming from the Uzawa algorithm and r being the parameter of the augmented Lagrangian.

Algorithm LAθ. A θ version of ***LA*** algorithm is elaborated, following an idea of [25] applied to the present variationnal problem. The equivalence theory between augmented Lagrangian and alternating-direction methods applied to the dual problem can lead to an augmented Lagrangian interpretation of the θ-scheme. Setting $\rho = r$, the resulting algorithm ***LAθ*** writes:

LAθ: (p^0, τ^0) *given. For* $n \geq 0$, $\theta \in [0; \frac{1}{2}]$, *we solve:*

$$2\eta_0 |d^n|^{s-2} d^n + r\theta d^n = r\theta \underline{\boldsymbol{D}}(u^n) + \tau^n \tag{98}$$

$$\left\{ \begin{array}{l} -r\theta \boldsymbol{D}^* \underline{\boldsymbol{D}}(u^{n+\theta}) - \nabla p^{n+\theta} = -f + \boldsymbol{D}^*(\tau^n) - r\theta \boldsymbol{D}^*(d^n) \\ \boldsymbol{div}(u^{n+\theta}) = 0 \end{array} \right. \tag{99}$$

$$\tau^{n+\theta} = \tau^n + r\theta(\underline{\boldsymbol{D}}(u^{n+\theta}) - d^n) \tag{100}$$

$$(2\eta_0 |d^{n+1-\theta}|^{s-2} + r(1-2\theta)) d^{n+1-\theta} = r(1-2\theta)\underline{\boldsymbol{D}}(u^{n+\theta}) + \tau^{n+\theta} \tag{101}$$

$$\tau^{n+1-\theta} = \tau^{n+\theta} + r(1-2\theta)(\underline{\boldsymbol{D}}(u^{n+\theta}) - d^{n+1-\theta}) \tag{102}$$

$$\left\{ \begin{array}{l} -r\theta \boldsymbol{D}^* \underline{\boldsymbol{D}}(u^{n+1}) - \nabla p^{n+1} = -f + \boldsymbol{D}^*(\tau^{n+1-\theta}) - r\theta \boldsymbol{D}^*(d^{n+1-\theta}) \\ \boldsymbol{div}(u^{n+1}) = 0 \end{array} \right. \tag{103}$$

$$\tau^{n+1} = \tau^{n+1-\theta} + r\theta(\underline{\boldsymbol{D}}(u^{n+1}) - d^{n+1-\theta}) \tag{104}$$

This algorithm ***LAθ*** solves equations (92) and (94) twice per iteration. Thus, in terms of computational time, ***LAθ*** is naturally twice as much expensive. Performances strongly depend on the values of θ and r and no convergence result exists yet. But, as we will see in Section 5 and Table 2, ***LAθ*** proves to be more efficient than ***LA*** for proper values of θ and r.

Let us point out that the derivation of both algorithms has been done in two dimensions but it naturally extends to the three-dimensional case.

4.5. Finite element discretization. Let γ and δ being two bilinear forms defined by:

$$\gamma(\mathbf{u}, \underline{\boldsymbol{d}}; \mathbf{u}^*, \underline{\boldsymbol{d}}^*) = \frac{r}{2} \left(\int_\Omega (\underline{\boldsymbol{d}} - \underline{\boldsymbol{D}}(\mathbf{u})) : (\underline{\boldsymbol{d}}^* - \underline{\boldsymbol{D}}(\mathbf{u}^*)) \, d\boldsymbol{x} + \int_\Omega \boldsymbol{div}(\mathbf{u}) \, \boldsymbol{div}(\mathbf{u}^*) \, d\boldsymbol{x} \right) \tag{105}$$

$$\delta(\mathbf{u}, \underline{\boldsymbol{d}}; p, \underline{\boldsymbol{\tau}}) = \int_\Omega \underline{\boldsymbol{\tau}} : (\underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{d}}) \, d\boldsymbol{x} - \int_\Omega p \, \boldsymbol{div}(\mathbf{u}) \, d\boldsymbol{x} \tag{106}$$

Then $(\mathbf{u}^*, \underline{\boldsymbol{d}}^*; p^*, \underline{\boldsymbol{\tau}}^*) \in (X \times H \times M \times H)$ is a saddle-point of $\mathcal{L}_r$ (defined by (91)) if and only if:

$$\left\{ \begin{array}{l} F(\underline{\boldsymbol{d}}) - F(\underline{\boldsymbol{d}}^*) + G(\mathbf{u}) - G(\mathbf{u}^*) + \gamma(\mathbf{u} - \mathbf{u}^*, \underline{\boldsymbol{d}} - \underline{\boldsymbol{d}}^*; \mathbf{u} + \mathbf{u}^*, \underline{\boldsymbol{d}} + \underline{\boldsymbol{d}}^*) \\ +\delta(\mathbf{u} - \mathbf{u}^*, \underline{\boldsymbol{d}} - \underline{\boldsymbol{d}}^*; p^*, \underline{\boldsymbol{\tau}}^*) \geq 0, \quad \forall (\mathbf{u}, \underline{\boldsymbol{d}}) \in X \times H \\ \\ \delta(\mathbf{u}^*, \underline{\boldsymbol{d}}^*; p, \underline{\boldsymbol{\tau}}) = 0, \quad \forall (p, \underline{\boldsymbol{\tau}}) \in M \times H \end{array} \right. \tag{107}$$

with $F(\underline{\boldsymbol{d}})$ and $G(\mathbf{u})$ defined respectively by (44) and (45).

We introduce the following finite dimensional spaces $X_h \subset X$, $H_h \subset H$, $M_h \subset M$. The discrete counterpart of the saddle-point problem then writes:

Find $(\mathbf{u}_h^*, \underline{\boldsymbol{d}}_h^*; p_h^*, \underline{\boldsymbol{\tau}}_h^*) \in (X_h \times D_h \times M_h \times T_h)$ such that:

$$\begin{cases} F(\underline{\boldsymbol{d}}_h) - F(\underline{\boldsymbol{d}}_h^*) + G(\mathbf{u}_h) - G(\mathbf{u}_h^* + \gamma(\mathbf{u}_h - \mathbf{u}_h^*, \underline{\boldsymbol{d}}_h - \underline{\boldsymbol{d}}_h^*; \mathbf{u}_h + \mathbf{u}_h^*, \underline{\boldsymbol{d}}_h + \underline{\boldsymbol{d}}_h^*) \\ +\delta(\mathbf{u}_h - \mathbf{u}_h^*, \underline{\boldsymbol{d}}_h - \underline{\boldsymbol{d}}_h^*; p_h^*, \underline{\boldsymbol{\tau}}_h^*) \geq 0, \ \forall (\mathbf{u}_h, \underline{\boldsymbol{d}}_h) \in X_h \times H_h \\ \\ \delta(\mathbf{u}_h^*, \underline{\boldsymbol{d}}_h^*; p_h, \underline{\boldsymbol{\tau}}_h) = 0 \ \forall (p_h, \underline{\boldsymbol{\tau}}_h) \in M_h \times H_h \end{cases} \tag{108}$$

It is required to have $\underline{\boldsymbol{D}}(X_h) \subset H_h$ in order for the bilinear form $\gamma(\mathbf{u}_h - \mathbf{u}_h^*, \underline{\boldsymbol{d}}_h - \underline{\boldsymbol{d}}_h^*; \mathbf{u}_h + \mathbf{u}_h^*, \underline{\boldsymbol{d}}_h + \underline{\boldsymbol{d}}_h^*)$ to vanish and consequently for the solution $\mathbf{u}_h$ not to depend upon the parameter $r > 0$. On the contrary, the resulting term of the augmentation would not vanish when the constraint $\underline{\boldsymbol{d}} = \underline{\boldsymbol{D}}(\mathbf{u})$ is satisfied.

No numerical analysis has been written for the present formulation. Let Ω_h be a triangulation of the domain Ω. We choose the Taylor-Hood spaces for the velocity-pressure pair $(\mathbf{u}, p)$:

$$X_h = \{\varphi \in X \cap \mathcal{C}^0(\bar{\Omega}), v_{|K} \in P_2(K)^2, \forall K \in \Omega_h\} \tag{109}$$

$$M_h = \{q \in M \cap \mathcal{C}^0(\bar{\Omega}), q_{|K} \in P_1(K), \forall K \in \Omega_h\} \tag{110}$$

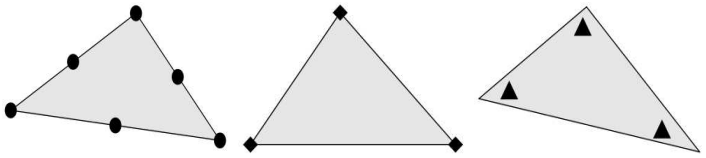
Next, we naturally define the following symmetrical tensor space:

$$H_h = \{\xi \in H, \xi_{|K} \in P_1(K)^3, \forall K \in \Omega_h\} \tag{111}$$

This triple extends the well-known *Taylor-Hood* finite element pair which is suitable for the classical two-field formulation of the power-law Stokes problem and includes the discontinuous piecewise linear function space H_h such that $\underline{\boldsymbol{D}}(X_h) \subset H_h$.

In other respect, as proved by [46] this triple uniquely defines (in the sense of inf-sup condition) the solution of a three-field formulation of the power-law Stokes problem first introduced in [8] in the case $1 < s < 2$.

We plot on Figure 2 the corresponding triangular finite elements triple.



(a) $\mathbf{u}_h \in (\mathbb{P}_2 - \mathcal{C}^0)^2$ (b) $p_h \in \mathbb{P}_1 - \mathcal{C}^0$ (c) $\underline{\boldsymbol{\tau}}_h$and $\underline{\boldsymbol{d}}_h \in \mathbb{P}_1^4$

FIGURE 2. *Extended Taylor-Hood mixed finite elements*

Again, the extension to the three-dimensional case is natural if one considers tetrahedral finite-elements; the corresponding finite element triple is identically refeered as the (P_2, P_1, P_1^{disc}) and corresponds respectively to the 9 nodes tetrahedron and the continuous and discontinuous 4 nodes ones.

We also recall that, using this discretization, all the boundary terms are applied to the velocity-pressure equation (92).

5. Performances of *LA* and *LAθ* Algorithms. This section is dedicated to the numerical assessment of ***LA*** and ***LAθ*** performances in terms of CPU time and memory consumption compared to the fixed point and the Newton methods. We use a sequence of increasingly finer (quasi-uniform) meshes for various power-law exponents. The meshes dimensions in terms of P_2, P_1 and P_1^{disc} degrees of freedom and the corresponding sizes for the Stokes system and the tensorial system are given in Table 1.

Comparisons are carried out on a bidimensional analytical test case and results for a realistic simulation with general boundary conditions are finally presented.

5.1. Convergence curves. The following solution $(\mathbf{u}^a, p^a)$ is considered on the unit square $\Omega = (0,1) \times (0,1)$:

$$\begin{pmatrix} u_x^a(x,z) \\ u_z^a(x,z) \\ p^a(x,z) \end{pmatrix} = \begin{pmatrix} \sin(m\pi x)^3 \sin(m\pi z)^2 \cos(m\pi z) \\ -\sin(m\pi x)^2 \sin(m\pi z)^3 \cos(m\pi x) \\ \sin(n\pi x) \sin(n\pi z) \end{pmatrix} \tag{112}$$

It is kinematically admissible ($\boldsymbol{div}(\mathbf{u}^a) = 0$). Then, the problem (22) is solved against $l(\varphi)$ defined by:

$$l(\varphi) = a(\eta(\mathbf{u}^a); \mathbf{u}^a, \varphi) + b(\varphi, p^a) \tag{113}$$

for the case $m = 1$ and $n = 2$ (see (112)).

Convergence curves are plotted on Figure 3 in order to validate the method and to obtain estimates for the orders of convergence.

From the explicit solution (112), one computes the error norms $E_{L^s}(\mathbf{u})$, $E_{W^{1,s}}(\mathbf{u})$, $E_{L^s}(\underline{\boldsymbol{d}})$, $E_{L^{s^*}}(p)$ and $E_{L^{s^*}}(\underline{\boldsymbol{\tau}})$ where:

$$E_{L^q}(\varphi) = \|\varphi - \varphi_h\|_{L^q} = \left(\int_\Omega |\varphi - \varphi_h|^q \right)^{1/q} \tag{114}$$

$$E_{W^{1,s}}(\mathbf{u}) = \|\mathbf{u} - \mathbf{u}_h\|_{W^{1,s}} = \left(\int_\Omega |\underline{\boldsymbol{D}}(\mathbf{u}) - \underline{\boldsymbol{D}}(\mathbf{u}_h)|^s \right)^{1/s} \tag{115}$$

These errors are plotted on Figure 3(b) for shear-thinning cases ($s = 1.16$ and $s = 1.33$), on Figure 3(a) for shear-thickening cases ($s = 2.25$ and $s = 3$) and on Figure 3(c) for the Newtonian case ($s = 2$). The reachable precision strongly varies with the value of s. The curve of $E_{L^s}(\underline{\boldsymbol{d}})$ also stands for $E_{W^{1,s}}(\mathbf{u})$ since both are extremely close for all $s > 1$.

In the Newtonian case, a convergence of order 2 is obtained for each unknown. The order 3 observed on $E_{L^s}(\mathbf{u})$ (the norm L^2 of the velocity error) is expected according to Aubin-Nitsche lemma applied to the Stokes problem on the convex unit square (see *e.g.* [17]). The order 2 is retrieved on the H^1 semi-norm. Errors on strain-rate tensor $E_{L^s}(\underline{\boldsymbol{d}})$ and deviatoric tensor $E_{L^{s^*}}(\underline{\boldsymbol{\tau}})$ strictly match since the viscosity is constant in this case and therefore $\underline{\boldsymbol{\tau}} = 2\eta_0 \underline{\boldsymbol{d}}$.

In the shear-thinning case, the order 3 for the norm L^2 on velocity does not occur but order 2 is maintained for $\mathbf{u}$ and $\underline{\boldsymbol{d}}$. The orders of convergence decline for p and $\underline{\boldsymbol{\tau}}$ when s tends to 1.

In the shear-thickening case, results are rather different. The error on the pressure $E_{L^{s^*}}(p)$ remains identical in both cases, providing an order of convergence of 2 for all s. The error $E_{L^{s^*}}(\underline{\boldsymbol{\tau}})$ increases with s but the order of convergence of 2 is also maintained. This very robust behavior seems to have consequences on the velocity approximation. The order of convergence obtained from $E_{L^s}(\mathbf{u})$ remains larger than 2 but the order of convergence evaluated from $E_{L^s}(\underline{\boldsymbol{d}})$ (which matches the one based on $E_{W^{1,s}}(\mathbf{u})$) significantly declines when s increases. It leads to quite high errors considering the P_2 quadratic approximation employed for the velocity discretization.

As a comparison, one can pay attention to the error bounds provided in [46] for the same discretization but applied to a *three-field* weak formulation of the power-law Stokes problem. In the shear-thinning case ($1 < s < 2$), finite-element errors for $\mathbf{u}$, p and $\underline{\boldsymbol{\tau}}$ with respect to the mesh step size h are bounded as follows:

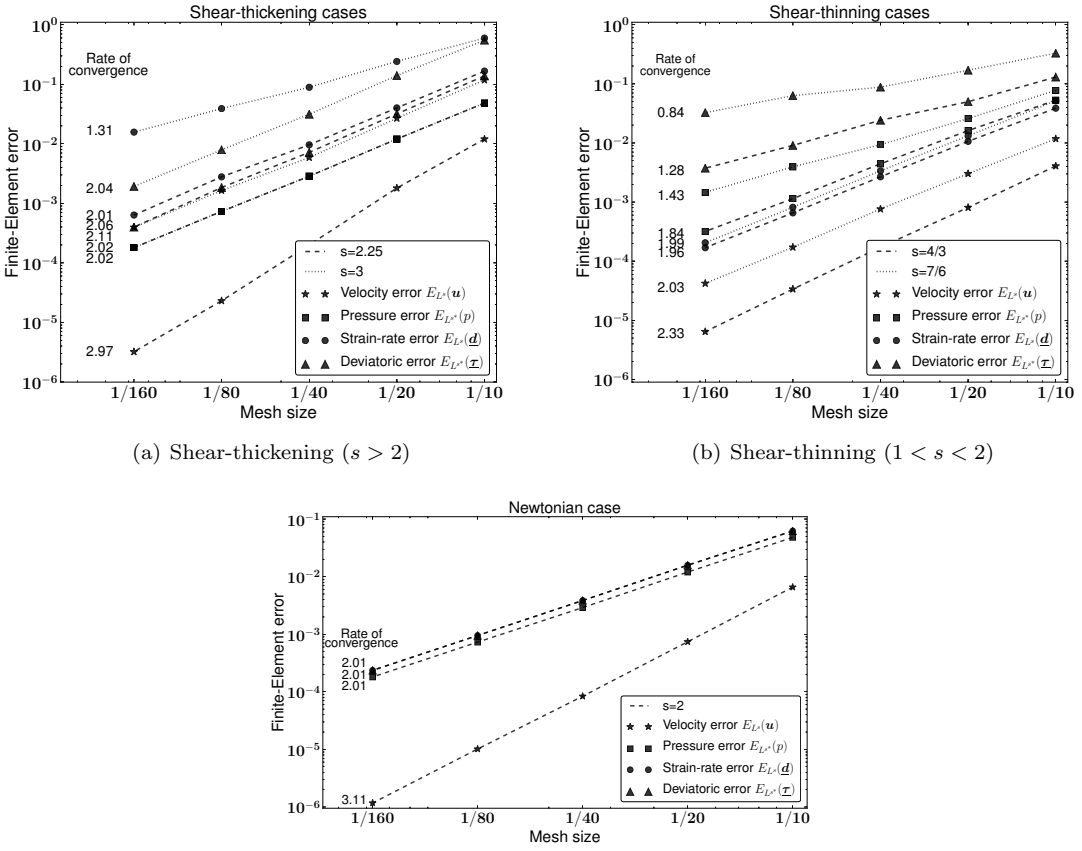
$$E_{L^{s^*}}(\underline{\boldsymbol{\tau}}) \leq Ch^{2(s-1)}, \tag{116}$$

$$E_{L^{s^*}}(p) \leq Ch^{2(s-1)}, \tag{117}$$

$$E_{W^{1,s}}(\mathbf{u}) \leq Ch^s \tag{118}$$

where $s^* = s/(s-1)$ is the conjugate exponent of s.

Our results match these estimations in the Newtonian case. For $s < 2$, the computed orders of convergence are significantly better than these estimations. Let us point out that no similar



(a) Shear-thickening ($s > 2$)

(b) Shear-thinning ($1 < s < 2$)

(c) Newtonian ($s = 2$)

FIGURE 3. *Convergence curves: variation with respect to the mesh-cell size h of the following finite element approximation errors (defined by* (114)*): $E_{L^s}(\mathbf{u})$(Velocity error), $E_{L^s}(\underline{\boldsymbol{d}})$ (strain-rate error), $E_{L^{s*}}(p)$ (pressure error) and $E_{L^{s*}}(\underline{\boldsymbol{\tau}})$ (deviatoric error). These errors have been computed using* ***LA*** *algorithm for various power-law exponent s ($s = 1.16, 1.33, 2.25$ and 3; exact solution given by* (112)*). The rates of convergence indicate the average logarithmic slope. Pressure error is identical for $s = 2.25$ and $s = 3$.*

estimate exist for $s > 2$. In the present four-field formulation discretized using the finite-element triple 2(a)-(2(b))-(2(c)) (see Figure 2), Figure 3 demonstrates the convergence of the method for all s.

We point out that the errors on velocity and pressure obtained from the Picard and the Newton algorithm are identical for $s < 3$. For $s \geq 3$ the fixed point method is unable to converge whereas Newton and both ***LA*** algorithms reach again the same accuracy for the converged state $(\mathbf{u}, p)$. The non-convergence of the Picard method could be due to the fact that the exponent $(s-2)$ in the power-law (7) becomes greater than 1 inducing a loss of contractivity for the functional.

5.2. Comparative performances. Table 2 indicates CPU time per iteration, global computational time and the number of iterations required for the Picard algorithm, the Newton-Raphson algorithm and the nonlinear augmented Lagrangian algorithms ***LA*** and ***LA$\boldsymbol{\theta}$*** to converged. The same four cases of nonlinearities are presented (*i.e.* $s = 1.16$, $1.33, 2.25$ and 3). The results have been obtained on the analytical solution (112) computed on increasingly finer unstructured quasi-uniform meshes (see Table 1). In the following, the Stokes system designates the linear system resulting from the assembly of the variational system (22) (or (92)). Its matrix M is square and belongs to $\mathcal{M}(2n_v+n_p)$ in $2D$ and to $\mathcal{M}(3n_v+n_p)$ in $3D$, n_v and n_p being the number of degrees of freedom for velocity and pressure (given in Table 1 in columns #dofv and #dofp respectively). The tensorial system designates the linear system associated to the variational equation (94) once the scalar quantity $|d|_{n+1}$ has been computed for each tensorial degree of freedom n_s (given in Table 1 in column #doft). Its matrix T is square and belongs to $\mathcal{M}(3n_s)$ in $2D$ and $\mathcal{M}(6n_s)$ in $3D$ since

Mesh	#cell	#dofv	#dofp	#doft	#StokesSP	#TensorSP	#memS	#memT
10×10	282	607	163	846	37923	2538	0.6	0.05
20×20	988	2057	535	2964	130544	8892	2	0.15
40×40	4258	8677	2210	12774	554988	38322	9	0.6
80×80	16866	34053	8594	50598	2200201	151794	35	2.5
160×160	67528	135697	34085	202584	8771837	607752	140	10
320×320	269972	541225	135627	809916	35129648	2429748	562	39
640×640	1084104	2170769	543333	3252312	141000088	9756936	2256	156

TABLE 1

Characteristic quantities of unit square meshes. The second column (#cell) indicates the number of triangles, the third column (#dofv) indicates the number of P_2 degrees of freedom discretizing the velocity, the fourth column (#dofp) the number of P_1 degrees of freedom for the pressure and the fifth column (#doft) the number of P_1^{disc} degrees of freedom for the tensorial unknowns $\underline{\underline{\boldsymbol{d}}}$ and $\underline{\underline{\boldsymbol{\tau}}}$. The sixth and seventh columns (#StokesSP and #TensorSP) report the number of non zero coefficients of the Stokes linear system of equation (92) *and the tensorial linear system of equation* (94)*. The eighth and ninth columns (#memS and #memT) represent an estimation of the memory, in megabytes, required to store the sparse Stokes and tensorial systems. The last column (#ratioM) is simply the ratio between #memS and #memT. From the system sizes and the number of non zero coefficients of the systems, one can deduce that the tensorial system is significantly sparser than the Stokes system.*

Picard algorithm									
s	#ite	**Mesh 80 × 80**		**Mesh 160 × 160**		**Mesh 320 × 320**		**Mesh 640 × 640**	
		#cpu	#time	#cpu	#time	#cpu	#time	#cpu	#time
1.16	54	2.3	126	17	920	125	6760	1100	59400
1.33	27	2.3	62	18	485	135	3650	1040	28100
2.25	11	2.3	25	17	190	140	1545	1130	12440
3	NA	2.4	NA	19	NA	145	NA	1140	NA
Newton algorithm									
s	#ite	**Mesh 80 × 80**		**Mesh 160 × 160**		**Mesh 320 × 320**		**Mesh 640 × 640**	
		#cpu	#time	#cpu	#time	#cpu	#time	#cpu	#time
1.16	4(+10)	3.3	37	22	265	155	2020	1130	26420
1.33	4(+3)	3.3	24	21	155	170	1260	1100	8610
2.25	4(+1)	3.2	15	21	100	180	870	1220	5980
3	8(+1)	3.2	28	20	200	160	1745	1190	13030
***LA* algorithm**									
s	r	#ite	**Mesh 80 × 80**		**Mesh 160 × 160**		**Mesh 320 × 320**		**Mesh 640 × 640**
			#cpu	#time	#cpu	#time	#cpu	#time	#cpu	#time
1.16	0.4	35	0.19	9.3	0.85	46	4.0	325	16	1830
1.33	0.45	22	0.18	6.4	0.88	36	4.3	270	16	1670
2.25	0.6	14	0.14	4.6	0.69	27	3.7	270	14	1450
3	0.4	26	0.14	6.2	0.68	35	3.6	305	13	1500
***LAθ* algorithm**									
s	r/θ	#ite	**Mesh 80 × 80**		**Mesh 160 × 160**		**Mesh 320 × 320**		**Mesh 640 × 640**
			#cpu	#time	#cpu	#time	#cpu	#time	#cpu	#time
1.16	1.5/0.2	13	0.39	7.5	1.8	40	8.6	290	31	1520
1.33	2/0.25	9	0.38	5.8	1.7	33	7.9	235	34	1360
2.25	3/0.4	6	0.29	4.3	1.4	26	6.8	215	29	1250
3	1/0.4	12	0.29	6.3	1.4	34	6.2	235	28	1490

TABLE 2

Comparative performances for Picard, Newton and LA algorithms for various power-law exponents s on the four finer meshes detailed in Table 1. Columns #ite indicate the number of iterations required for the algorithm to reach a 10^{-5} converged velocity. For the Newton algorithm, the number in brackets gives the minimum number of Picard iterations required to compute the first-guess. Columns #cpu indicates the approximate time, in seconds, required to perform **one** *iteration. Column #time indicates the total computational time (except the pre-processing time, for instance the reading of the mesh). Columns r (and θ) report the corresponding values used for the numerical parameters of* $\boldsymbol{LA}$ *(and* $\boldsymbol{LA\theta}$*) algorithm. The computations are sequential and the estimated times have been obtained with a* $2.6Ghz$ *core.*

it is symmetrical. The computation stops when the relative variation between two successively computed velocity fields is lower than 10^{-5}. The number of iterations and consequently the CPU time required for $\boldsymbol{LA}$ algorithms depends on the value of the augmented Lagrangian parameter r (and θ for $\boldsymbol{LA\theta}$). The results given in Table 2 indicate the best observed performances and the corresponding values for the parameters r and θ are provided.

The values of s "close" to 2 (here $s = 2.25$ and $s = 1.33$) represent *close-to-linear* situations.

In these situations, Picard algorithm and $\boldsymbol{LA}$ algorithms require approximatively the same number of iterations. Algorithm $\boldsymbol{LA\theta}$ solves twice the same equations as $\boldsymbol{LA}$ and numerical experiments show approximatively half as many iterations. When s goes away from 2, the trend changes and both algorithms $\boldsymbol{LA}$ and $\boldsymbol{LA\theta}$ give better performances in terms of iterations compared to the fixed point. For proper values of r and θ, $\boldsymbol{LA\theta}$ can provide slight improvements, particularly in terms of *iterative time* (*i.e.* the time spent in the nonlinear loop which is equal to (#ite×#cpu) as defined in Table 2).

The Newton method is able to converge with a high rate of convergence for every values of s but when s tends to 1, the functional (36) degenerates and it requires many pre iterations (from the Picard algorithm for example) to reach the basin of attraction. Typically, for $s < 1.5$, the Newton method has to be initialized with a state $(\mathbf{u}_0, p_0)$ such that $\|u_0 - u^*\|/\|u^*\| \sim 0.1$. On the contrary, the Newton algorithm is quite robust for $s > 2$ and does not require an almost converged first-guess. However, for $1 < s < 2$, the fact that the Newton system (36) is singular for $u_k = 0$ can represent a severe drawback for the use of this method. Indeed, the rate of convergence of the Newton algorithm applied to the analytical test case (112) in the case of a small average velocity strongly deteriorates. If the velocity norm is small enough (*e.g.* 10^{-8} instead of 1 in the present case), the Newton algorithm is unable to converge. This property is not desirable when one solves low Reynolds macroscopic flows.

One iteration of the Newton method is slightly longer than a fixed point iteration due to the assembly of a Stokes system and a matrix-vector product performed at each iteration in order to compute the right-hand side.

In all situations, the principal interest lies in the computational time. Regarding the time per iteration, the ratios between Picard or Newton algorithms and $\boldsymbol{LA}$ are very large and increase with the size of the mesh.

First, for Picard and Newton algorithm, the total computational time is approximatively equal to the time per iteration multiplied by the number of iterations. This computational time, as well as the scaling with increasingly finer meshes, is thus strongly depending on the linear solver considered (which is the multifrontal direct solver MUMPS, see [3]). The present experiments show a computational time growing approximatively with $(\#\text{cell})^{3/2}$: for a mesh with four times as many cells, the time is multiplied by 7 to 8. This is an excellent ratio for a direct solver as it is smaller than $(\#\text{cell})^2$.

For $\boldsymbol{LA}$ algorithm, the total computational time is much bigger than the *iterative time* (the time spent in the nonlinear loop) and the ratio between both increase with increasingly finer meshes. On the 80×80 mesh, the iterative time represents between 40 to 70% of the total time whereas, on the 640×640 mesh, it ranges between 15 and 30%. This behavior is explained by the fact that the solve of the tensorial system is extremely fast due to its specific matrix structure and the fact that the solve of the Stokes-like system defined by (92) is almost instantaneous since the matrix has been pre-factorized before the nonlinear loop. These two properties are preserved with increasingly finer meshes. The Stokes solution remains negligible and the tensorial system solution time grows approximatively with #cell , the number of cells (or similarly the number of degrees of freedom or the number of non-zero values in the tensorial system, see Table 1). The details are given in the next paragraph.

Iterative time. For both Newton and Picard algorithms, the assembly of the Stokes system M corresponding to the new viscosity field has to be computed at each nonlinear iteration. On the contrary, in $\boldsymbol{LA}$ algorithm, the nonlinear part of the problem appears in another equation, and then the Stokes type matrix M (defined by equation (92)) does not contain a viscosity changing in every nonlinear iteration. Thus it can be computed and factorized once and for all before the algorithm starts. The cost of this step (the Stokes solution) is thus negligible within the nonlinear loop, even for large matrix system.

The cost for one iteration comes essentially from the time required to solve the tensorial nonlinear system. The nonlinear equation (94) requires first to compute the norm $|d|_{n+1}$ of the rate of deformation tensor d_{n+1} using a Newton algorithm on the n_s norm values. This step as a rather small cost but non negligible given the very small time per iteration. This step is the one that brings a noticeable difference between the times per iteration for the different values of s (for a given mesh). It clearly appears that this Newton step is faster for shear-thickening situations ($s > 2$).

The tensorial matrix T contains approximatively one fifth as many non-zero coefficient (see Table 1, column #ratioM) making it lighter to assemble, store and solve. In addition, the tensorial system, when discretized using a discontinuous Galerkin approximation, shows an extremely convenient structure which can be exploited. As a matter of fact, the resulting matrix is block diagonal at element level. Then solving the system amounts to solving a sequence of 3×3 linear systems in $2D$ and 9×9 linear systems in $3D$. These solutions can be done explicitly along the assembly of the elementary matrices. This property has two major consequences. The first one is the time saving since such a resolution becomes relatively costless (no assembly and no factorization are required). Since only the elementary systems are solved, the computational time grows with the number of cells #cell which provides an excellent scalability for the time per iteration. From Table 2, we can see that the ratio is a little bigger than *#cell* (typically around 5 instead of 4 for a mesh with four times as many cells). This difference is due to the inner Newton algorithm performed to obtain $|d|_{n+1}$ at each nonlinear iteration.

It is interesting to notice that this particular structure makes the solution of the tensorial system an easily parallelizable step since each elementary system is independent from the others and no domain decomposition is required.

Memory. The other resulting feature is the important memory saving one can obtain in a data assimilation context. As pointed out above, the adjoint state is obtained from reverse accumulation which requires to store each state encountered throughout the iterative loop. For Picard and Newton method (or conjunction of both) the resulting memory consumption corresponds to the required memory to store one Stokes system multiplied by the number of iterations performed by the algorithm (see [39]). Estimations of the storage of a sparse Stokes system are given in Table 1, column #memS. Irrespective of the discretization, the sparse tensorial system is significantly smaller to store (see Table 1, columns #memT and #ratioM). In the present situation, the structure of the tensorial system allows to solve it along the assembly. Consequently, no storage of this matrix is needed and the memory consumption of $\boldsymbol{LA}$ algorithm reduces to the storage of one sparse Stokes system. Typically, for $s = 1.33$ it leads to a memory consumption ratio of 27 compared to the use of a fixed point algorithm. The gain grows with the nonlinearity.

5.3. Summary. From all these observations and Table 2, we can globally state that, for well chosen values of the parameters r (and θ) that provide a good rate of convergence (and thus a limited number of iterations), the total time of computation using $\boldsymbol{LA}$ algorithm converges to the time required for factorizing and solving *one Stokes system* as the mesh grows. As a matter of fact, since the *iterative time* using $\boldsymbol{LA}$ grows approximatively with #cell, the number of cells of the mesh (or equivalently with the number of degrees of freedom), the step of factorization of the Stokes system becomes dominant on larger mesh.

Algorithm $\boldsymbol{LA\theta}$ provides a good convergence speed (*i.e.* the number of iterations required to reach a given threshold) for well-chosen values of parameters r and θ and slightly improves algorithm $\boldsymbol{LA}$. If one takes into account the pre-iterations required for the Newton method to be initialized, the convergence speeds are similar for $\boldsymbol{LA\theta}$ and Newton-Raphson for all values of s (see Table 2). However, the negligible time of the nonlinear loop for large meshes reduces the interest of $\boldsymbol{LA\theta}$ because of the introduction of a second tunable parameter.

The very good scalability of $\boldsymbol{LA}$ makes it very well suited for large domains with a great

number of elements. The time ratio compared to Newton or Picard, for sufficiently large mesh, converges to the number of iterations required for Picard or Newton to converge.

It is interesting to notice that the best values for parameters r (and θ) remain the same when the mesh grows. It follows that, in the abscence of a rule to choose these parameters, an approach could be to tune them on a coarser mesh before using them to run a larger simulation.

In other respect, the various issues of the Newton-Raphson algorithm for the present power-law Stokes problem (namely the ill-posedness of the principal operator for a constant u_0, the degeneration of the bilinear form for small value of s and especially the small radius of its convergence disk) makes it an unreliable choice as a solver for realistic simulations and a fortiori for an identification parameter process for pseudoplastic fluids. For the simulations of dilatant fluids, the functional (36) is continuous and does not degenerate and Newton algorithm is suitable for this type of fluid.

The computational times assessed in this study depend on the efficiency of the linear solver. We observed a time increase with the number of degree of freedom close $\#\text{cell}^{3/2}$ for the direct linear solver used here. If a faster solver is considered, the time saving ratios could be different.

In the case of the bidimensional glaciological simulation described in Section 7.3, the use of $\boldsymbol{LA\theta}$ for the forward problem allows to divide the time of computation by 6 compared to the fixed point (for a 10^{-5} converged velocity with $r = 1.8$ and $\theta = 0.2$). Indeed, the same run using Picard algorithm results in 24 iterations and 22 seconds (instead of 16 iterations and 3.7 seconds for $\boldsymbol{LA\theta}$) to converge. In terms of memory, the computation of the adjoint state using $\boldsymbol{LA\theta}$ would then require the storing of 1 Stokes sytem instead of 24 when using the fixed point.

6. Local sensitivity analysis. Sensitivity analysis allows to study how a perturbation on the input of a model induces perturbations on its response. The local sensitivity, based on the Taylor expansion, analyses the behavior of a system in the neighbourhood of a given state for a static system or of a given trajectory for a dynamic system. The use of the adjoint method to perform sensitivity analysis provides the directional Fréchet-derivative of a functional with respect to all its parameters with the integration of one extra model only (unlike the direct approach which brings a sensitivity with respect to one parameter for a given direction). Then, the adjoint model is well suited to study systems with a large number of parameters and/or spatially distributed parameters.

The calculation of local sensitivities contributes to improve the understanding of the system by identifying important parameters, to determine the influence of parameters variations on the behavior of the system and eventually to improve the understanding of the physics. Hereafter, we will focus on free-surface problem at a given time (*i.e.* for a domain Ω fixed) thus providing *quasi-static* sensitivities. Adjoint method with a moving free-surface would require extra terms due to the domain shape modification, which are not addressed in the present study.

Originally, DassFlow (*Data assimilation for free-surface Flows*) software implements shallow water equations (see [35] and [32]). We extended it to the present nonlinear Stokes problem. It is designed for variational data assimilation (classicaly called 4D-var method). Variational data assimilation is based on the optimal control theory and the adjoint equations (see *e.g.* [37]). Briefly, this method consists in computing a control vector value $\boldsymbol{k}$ minimizing a cost function which measures the discrepancy between the computed variable and available data. In our case the control vector $\boldsymbol{k}$ includes spatially distributed variable such as inflow and outflow, lateral stresses, the friction coefficient β, the rheological constant η_0 and the rheological exponent s.

6.1. Observations and cost function. We call $\boldsymbol{k_0}$ the control variables which define the true state, a priori unknown, in which observations u^{obs} have been generated. We will consider two types of observations in the following. If one observes the normal profile of a streamwise velocity of the flow at a given abscissa x_0, we have:

$$u^{obs} = u^{obs}(\boldsymbol{k_0}; x_0, z),\ b \leq z \leq H \tag{119}$$

where b stands for the ordinate at bottom and H the ordinate at surface. In the case of observed surface velocities, we have: $u^{obs} = u^{obs}(\boldsymbol{k_0}; x, H)$. Given velocity observations denoted u^{obs}, we define the cost function j which measures the discrepancy between the computed variable (state of the system) and the available data as follows:

$$j(\boldsymbol{k}) = \frac{1}{2}\int_X \left|\mathbf{u}(\boldsymbol{k}, z) - u^{obs}\right|^2 \, dx \, dz \tag{120}$$

where $X = \Gamma_s$ in the case of an observed velocity profile and $X = [0, h]$, h being the thickness of the domain at $x = x_0$, in the case of observed surface velocities.

6.2. Adjoint model. In order to compute efficiently all partial derivatives of the cost function $j(\boldsymbol{k})$ with respect to the components of the control vector $\boldsymbol{k}$, we introduce the adjoint model (see *e.g.* [37]). As we saw previously, in DassFlow software, we use the automatic differentiation of the source code to obtain the adjoint model (see [32]). This last approach ensures a good consistency between the computed cost function (including all types of errors -errors of discretization, rounding errors, iterative algorithms, etc.) and its gradient since it is the computed cost function which is differentiated. In the case of DassFlow, the direct code is written in Fortran 95 and it is derived using the automatic differentiation tool Tapenade (see [30]). In order to obtain the "computational gradient" (partial derivatives of the cost function j using differentiation of the forward code), the procedure consists in running the direct code then the adjoint code. We point out that j must be included into the response of the forward code.

We recall that a single integration of the direct model (15)-(16) followed by a single integration of the adjoint model allow to compute all components of the gradient of the cost function. We address to [32] for more details.

7. Numerical experiments. This section presents the numerical experiments and the corresponding sensitivity analyses. The first one is based on a laboratory experiment at small scale described in [12]. It consists in a gravity-driven surge of kaolin on an inclined plane considered at steady state (*i.e.* $\mathbf{u} \cdot \boldsymbol{n}|_{\Gamma_s} = 0$). For this flow we focus on the sensitivity with respect to the spatially distributed power-law index s (see equation (7)) considering an observed velocity profile at a given abscissa (see Figure 4). The second one is a glaciological application at large scale that uses radar-sensed profiles of the surface and the bottom of a continental glacier in Antarctica due to [36]. According to current glaciological questions, we study the sensitivity of this flow with respect to the friction coefficient (see equation (13)) distributed on the boundary Γ_b. We consider surface velocities as observations. In both cases, data are synthetic (*i.e.* numerically generated).

These flows being steady and relatively shallow, we first present the analytical solution of the uniform steady power-law Stokes problem with friction condition at bottom. The goal is to use it as an inflow Dirichlet condition for both simulations. This analytical solution is not classical in the presence of basal friction.

7.1. Analytical Poiseuille-like solution. We call *permanent flow in a channel*, the steady ($\partial_t \cdot = 0$) uniform ($\partial_x \cdot = 0$) solution of the 2D viscoplastic Stokes problem on a given domain Ω corresponding to a channel with a flat bottom and a constant thickness $h = (H - b)$ with an angle of inclination θ. We consider a friction condition at bottom (defined by (13)-(14)). The stationary free-surface condition on the upper surface corresponds to $\sigma \boldsymbol{n}_{|\Gamma_s} = 0$ and $u \cdot n_{|\Gamma_s} = 0$. From equations (15)-(16), we calculate the expression of the solution $\mathbf{u} = (u_\tau, u_n)^T$ of this flow in the "mean-slope" coordinate system $\bar{\boldsymbol{x}} = (\bar{x}, \bar{z})$ with $\bar{x} \in [0, L]$ and $\bar{z} \in [0, h]$. We obtain the following expression for the tangential and normal component of the velocity:

$$u_\tau(\bar{z}) = \frac{-\rho g \sin(\theta) h}{\beta} + \frac{s-1}{s}(2\eta_0)^{\frac{1}{1-s}}(\rho g \sin(\theta))^{\frac{s}{s-1}}(h^{\frac{s}{s-1}} - (H - \bar{z})^{\frac{s}{s-1}}) \ , \ u_n = 0 \tag{121}$$

7.2. The viscoplastic steady wave.

7.2.1. Description. Following [12], we present hereafter the test case of a steady wave of non-Newtonian fluid coming from a gravity-driven surge on a inclined plane of inclination θ. We will consider the rheological parameters of the *kaolin slurry* (refined clay) used in their experimental setup but we focus exclusively on the pseudoplastic behavior of kaolin slurry and thus, modeling it using a power-law representation (thus neglecting the *Herschel-Bulkley* yield stress and time dependent-process such as viscoelastic behavior). The inclined plane is in fact a conveyor belt moving at constant velocity $u_\tau^b = \mathbf{u} \cdot \boldsymbol{\tau}_{|\Gamma_b}$ in the upstream direction toward a fixed wall. Then a given volume of kaolin slurry, abandoned on the conveyor belt converges to a front shape whose dimensions depend on the inclination θ and the velocity of the belt u_τ^b. The following test case is based on the results presented in [12] and we address this paper for more details on the experimental setup.

In such a flow, three zones can be schematically distinguished: the front, a region where flow height is uniform, and a region influenced by the upper wall. The upstream zone perturbed by the wall is of relatively short length and appears to remains confined (see [12]). Then, neglecting the upper wall, the flow far from the front corresponds to the permanent flow introduced in (121) and thus, this velocity profile can be used as an inflow for the numerical simulation. Moreover, as noted in [12], slipping at the bottom is negligible. Hence, we will consider an homogeneous Dirichlet condition. The free upper-surface is subject to free-constraint condition. The physical parameters used for the simulation are presented in Table 3. The model is the incompressible Stokes problem (15)-(16) solved on the geometry plotted on Figure 5. Input parameters as well as the shape of the domain are determined using the experimental measurements presented in [12].

The mesh is triangular and made of $N_{cell} = 23744$ cells. We plot on Figure 4 the input velocity profile computed from (121) and the computed streamwise velocity profile in the uniform region (here at $x_0 \sim -16cm$) with respect to the normal thickness $\bar{z}$. Then, the Reynolds number at which the flow occurs is:

$$\mathcal{R}e = \frac{\rho U^* H^*}{\eta_e} = \frac{\rho U^* H^*}{\eta_0 \left(\frac{U^*}{H^*}\right)^{s-2}} \sim 10^{-1} \tag{122}$$

and low Reynolds approximation is valid. It follows that the Stokes problem is suited for the simulation of this flow.

As noted in [12], rheological parameter measurements are possible but with large uncertainties (see also [4]). As a first step, we perform sensitivity analysis with respect to a locally defined rheological exponent s to highlight area sensitive to the rheological behavior using the cost function (120). We take as observations u^{obs} the computed vertical velocity profile at x_0 plotted on Figure 4.

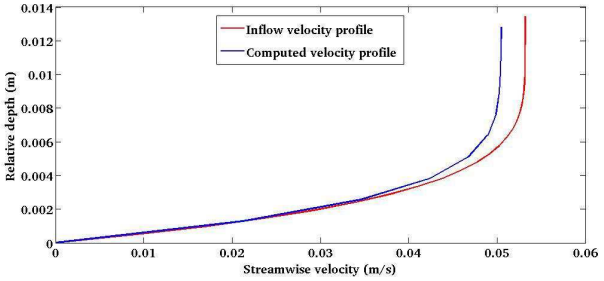


FIGURE 4. *Computed velocity in the surge at x_0 and inflow velocity at $x = 0$*

7.2.2. Numerical results. We present on Figure 5 the sensitivity of the model with respect to the rheological exponent s. In this case, s is locally defined (constant by cell) and consequently $\partial_s j$ has the same dimension as s.

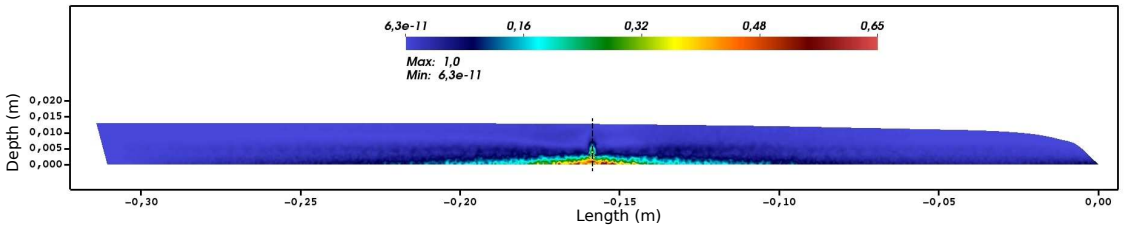


FIGURE 5. *Kaolin wave: Normalized sensitivity with respect to the spatially distributed power-law exponent* $\partial j/\partial s(s_0) \in \mathbb{R}^{N_{cell}}$, $s_0 = 2$ *in the mean-slope frame. The velocity profile plotted on Figure 4 (obtained with* $s = 1.24$*) is used as synthetic data. The scale does not reach* 1 *because cell values have been interpolated on the vertices of the mesh.*

First, from Figure 4, one can state the uniformity of the flow in the region "far from the front". The slight decreasing of thickness between the inflow boundary and the observation zone at x_0 explains the small differences between the two vertical velocity profiles.

Figure 5 shows that the normalized sensitivity to the local power-law exponent s is high close to the line of observation (represented with a dash line). Besides, it clearly appears that the highest sensitivity region is concentrated close to the bottom. The sensitivity at upper surface is close to zero. Hence a precise knowledge of the power-law index s close to surface would be quite useless to reproduce the observed velocities u^{obs}. It follows that the sensitivity is correlated to the shearing in the flow, which is maximum at bottom and goes to zero at free-surface. Then we can state that, in terms of rheology, the computed velocity at the surface depends almost only on the rheological behavior *at bottom where the shearing is the stronger*. This result is all the more remarkable since the misfit between computed and observed velocities is greater at surface and close to zero at bottom (see Figure 4). As noted in [12], rheometer measurements are spoiled by experimental artifacts for small and large shear rates. Then, from this sensitivity analysis, one can state that the behavior at high shear rates is of primary interest to constrain the model and rheological measurements should be done in this perspective. Furthermore this result allows to state that, in this quasi-uniform flow situation, the infinite (singular) viscosity predicted by the power-law at surface (for $1 < s < 2$, see equation (29)) does not affect the computation in terms of surface velocities.

7.3. Geophysical test case: the Mertz glacier.

7.3.1. Description. We compute hereafter an ice flow subject to a gravity source term over a sloppy bedrock. This type of flow are extremely viscous and slow and then occurs at low Reynolds. They are observed, at "small scale", in *mountain glacier* such as in Himalaya, Andes, Alps, etc. ($\mathcal{R}e \sim 10^{-13}$) and at larger scale in *ice-sheets*: Greenland and Antarctica ($\mathcal{R}e \sim 10^{-10}$) (see *e.g.* [28]). The flow of ice, if considered isotropic, is described by an empirical constitutive law called *Glen's law*, corresponding to a power-law and brought to light in [23].

In order to better forecast current climatic trends (typically in terms of sea-level rise), an important issue is to obtain accurate simulations of glaciers hence to calibrate the models. The observations potentially available in this context are: the topography of the bedrock, the surface elevation and surface velocities. One of the main uncertainties in glaciological modelling is the friction coefficient β. It is a quantity which link the sliding speed $(\mathbf{u} \cdot \boldsymbol{\tau})|_{\Gamma_{fr}}$ to the basal shear-stress $\sigma_{n\tau}$. The physical meaning of β is unclear since it can represent many processes such as *e.g.* the roughness of the bedrock, the basal hydrostatic pressure of subglacial water cavity, the rheology of a subglacial sediment boundary layer and more generally the thermal conditions on the bedrock (see *e.g.* [28]). Since this quantity is unmeasurable, the use of control methods to infer β is of prime interest to calibrate ice flows. As a first step toward data assimilation, sensitivity analysis helps to better understand the underlying physics of the modelling of sliding and quantify the role of the sliding coefficient in terms of model response.

The computational domain is built from real field data; topography of the bedrock and of the

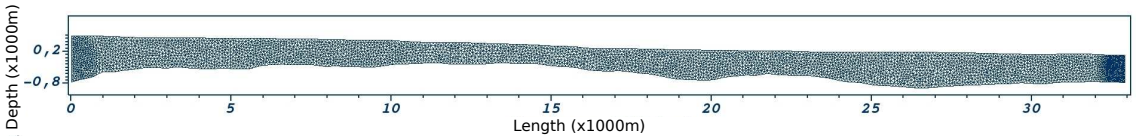


FIGURE 6. *Vertical cut of the outlet glacier Mertz, Antarctica (topography profile from [36]) , x-scale* $= 2/5$

Rheological exponent	*Average slope*	*Gravity*	*Fluid viscosity coefficient*	*Kaolin slurry density*
$s = 1.24$	$\theta = 15°$	$g = 9.81m/s^2$	$\eta_0 = 17.04Pa \cdot s^{0.24}$	$\rho = 1399kg/m^3$

TABLE 3
Physical input parameters of the kaolin slurry surge

surface are bidimensional radar-sensed layer of the *Mertz glacier* in East Antarctica. These layers have been measured along a flowline of this glacier (Italian Antarctic Expedition 2000, see [36]). Our study focuses on the grounded part of the glacier, called the outlet glacier. The domain is represented on Figure 6 with an horizontal scale of 2/5. The mesh is triangular and made of 8241 cells. The model is the incompressible Stokes problem (15)-(16) at steady state solved on the real geometry with the following boundary conditions: free-constraint on the upper surface Γ_s, friction condition on the bedrock Γ_{fr} (see equation (13)-(14)), prescribed horizontal velocity profile $u_l(z)$ on the vertical inflow boundary Γ_l, hydrostatic pressure of ice on the outflow boundary Γ_r where $h(z)$ represents the height of ice as a function of the ordinate z. The left boundary is subject to a non homogenous Dirichlet condition on the velocity. The prescribed streamwise velocity $u_\tau(z)$ is given by the analytical Poiseuille-like flow solution (121) using parameters given in Table 3. The right boundary is subject to a given normal stress that corresponds to hydrostatic pressure for the permanent flow, that is to say : $\underline{\boldsymbol{\sigma}}\boldsymbol{n}(z) = \rho \boldsymbol{g}(h_s - h(z))$ where h_s represents the height of the surface at the right boundary. In order to be consistent with the encountered velocity magnitudes, runs are performed using a $(MPa - m - a)$ unit system. The physical parameters used for the runs are presented in Table 4 (see *e.g.* [28]).

7.3.2. Numerical results. The cost function used for these sensitivity analysis is :

$$j(\beta, \eta_0) = \int_{\Gamma_s} \|u_s^{obs} - u_s(\beta, \eta_0)\|_2^2 \ \ d\boldsymbol{x} \tag{123}$$

The target surface velocities u^{obs} are perturbated with a random white noise of 1%. They are obtained using a linear sliding law and a spatially varying sliding coefficient defined by :

$$\beta(x) = \begin{cases} 0 \text{ if } x \in [x_i, x_f] \\ 1.5 \cdot 10^{-3} \text{ otherwise} \end{cases} \tag{124}$$

where the free-slip zone $[x_i, x_f]$ could represent a subglacial lake or more generally an area of perfectly lubricated interface. Here we take $x_i = 2.5 \cdot 10^3$m and $x_f = 2.7 \cdot 10^3$m. The sliding coefficient is defined constant by edge on the bottom. Thus, the corresponding sensitivity $\partial j / \partial \beta(\beta_0)$ is also a function of the abscissa. Synthetic surface velocity observations u_c^{obs} and u_f^{obs}, obtained with a constant $\beta = 1.5 \cdot 10^{-3}$ and with β defined by (124) respectively, are plotted on Figure 7 at the value $\beta_0 = 0.01$

Rheological exponent	*Average slope*	*Gravity*	*Ice density*
$s = 4/3$	$\theta = 2 \cdot 10^{-2}\ rad$	$g = 9.756234 \cdot 10^{15}\ m \cdot a^{-2}$	$\rho = 9.1376 \cdot 10^{-19}\ MPa \cdot a^2 \cdot m^{-2}$
Fluid viscosity coefficient	*Fluid height of the uniform flow*	*Friction coefficient*	
$\eta_0 = 40\ MPa \cdot a^{\frac{1}{3}}$	$h = 916.21\ m$	$\beta = 1.5 \cdot 10^{-4}\ MPa \cdot a \cdot m^{-1}$	

TABLE 4
Physical input parameters of the Mertz glacier

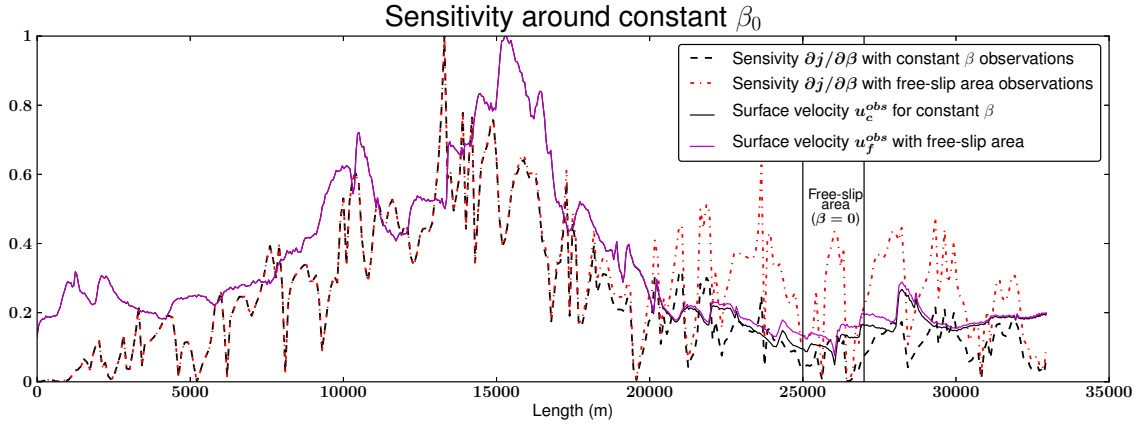


FIGURE 7. *Plain lines: computed surface velocities u_f^{obs} obtained using the friction coefficient defined by* (124) *(in purple) and u_c^{obs} obtained using a constant value $\beta_c = 1.5 \cdot 10^{-3}$ (in black) rescaled to remain smaller or equal to* 1. *Dash lines: computed gradient $\partial j/\partial\beta$ using u_f^{obs} as synthetic data (in red) and $\partial j/\partial\beta$ using u_c^{obs} as synthetic data (in black)*

A first interpretation is that the sensitivity follows the variations of surface velocity; a high velocity corresponds to a high sensitivity of the model to β. As noted in [40], it seems quite natural since a higher velocity means a larger misfit and consequently a larger gradient.

This high correlation suggests that surface velocities are well-suited as observations to identify the sliding coefficient.

However, Figure 7 shows an area of large sensitivity between $x = 20$km and $x = 30$km which does not corresponds to a specifically high velocity zone. The steep vanishing of β between $x_i = 25$km and $x_f = 27$km induces a large gradient despite a smaller misfit in and around the free-slip area. This "singularity" in the sliding coefficient was not obvious from the variations of surface velocities around the free-slip event.

One of the main process at stake in this rapid ice-stream type flow is the fluctuation of the pressure of subglacial meltwater inducing a "stick-slip" phenomenon (see [22]). This quasi-static sensitivity allows to detect at a given time the presence of a momentary increase in sliding rate given a geometry and surface velocities without transient simulation. The history and variations of the sliding rate can then be reconstructed through the quasi-static sensitivities and the fluctuations can be traced back (from a correct topography and surface measurements). In other respects, these sentivities can also help defining areas of interest for sliding measurements since the fluctuations in water pressure for basal cavities are local in space and time.

As underlined in [28], "the transmission of basal movement to the surface is non-local and is strongly filtered". Here, both these phenomena are clearly highlighted and the sensitivity analysis tool demonstrates its ability to help the understanding of the physics by providing *a priori* interpretations of the observed flow; the gradients of the fluid model help to see beyond the filtering and the non-local behaviour.

8. Concluding remarks. This work presents a three-field finite element solver for quasi-Newtonian flows based on a four-field saddle-point formlation. The well-posedness of the continuous problem associated with this four-field formulation was studied. The existence of a solution to the corresponding saddle-point problem is proved. The methodology is based on an augmented

Lagrangian formulation and can then be extended to the solution of the non differentiable elastic-viscoplastic flow problem. A comparison of the solver with the fixed point and the Newton method demonstrates the strong numerical performances of the present algorithms $\boldsymbol{LA}$ and $\boldsymbol{LA\theta}$; the computational time computation ratio between the Newton method and $\boldsymbol{LA\theta}$ ranges between 5 and 15 for a one million element mesh along with low memory consumption. Algorithms $\boldsymbol{LA}$ and $\boldsymbol{LA\theta}$ also show good robustness in a larger range of power-law exponent than both fixed point and Newton method.

Using adjoint method, we performed sensitivity analysis on the nonlinear Stokes problem. In order to be coherent with physical problems, numerical experiments has been done on realistic test cases of current power-law flow problems. The experimental surge of kaolin slurry shows high sensitivity to the rheological exponent close to the bottom *i.e.* in the highly sheared zone establishing the importance of accurate rheological measurements at high-shear rate to constrain the flow. The flow of the Mertz outlet glacier in Antarctica also demonstrates the ability for the sensitivity analyses to detect a high frequency event in the basal sliding and more generally the strong modeling role of this parameter. As pointed out by Vaughan and Arthern [47], the modeling of the ice base is a crucial step to obtain accurate simulations and remains poorly understood and constrained. The simple numerical experiment presented here shows that the wide range of processes at stake in the basal slip and their coupling can be studied and quantified using sensitivity analysis.

In a parameter identification perspective, the $\boldsymbol{LA\theta}$ algorithm which provides faster calculations along with low memory consumption is of prime interest. On the one hand, the computation cost of a data assimilation cycle, in terms of CPU time, is estimated around *50-100 times the cost of a direct run* (when using a first order descent algorithm). In addition the accuracy of the gradient computed by the adjoint code is strongly linked to the accuracy of the direct solver. Then improving the direct run could lead to important time saving for data assimilation. On the other hand, the reverse accumulation technique is significantly memory consuming and the present implementation of $\boldsymbol{LA}$ algorithm represents a strong asset to overcome this problem.

Acknowledgments. This work was partially supported by PRES Toulouse, with the PhD funding of the first author. We want to thank Benoit Legrésy (LEGOS) for his data on the Mertz glacier and Ronan Madec (IMT) for his help on the development of the adjoint code. This work was also supported by *Agence Nationale de la Recherche* through ADAGe project No. ANR-09-SYSC-001. We also thank INSA students for developments within the software.

REFERENCES

[1] R. A. Adams and J. J. F. Fournier, *Sobolev Spaces*, Academic Press, July 2003.
[2] G. Allaire, *Numerical analysis and optimization*, Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, (2007).
[3] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 15–41.
[4] C. Ancey, *Plasticity and geophysical flows: A review*, Journal of Non-Newtonian Fluid Mechanics, 142 (2007), pp. 4–35.
[5] C. Ancey and S. Cochard, *The dam-break problem for Herschel-Bulkley viscoplastic fluids down steep flumes*, Journal of Non-Newtonian Fluid Mechanics, 158 (2009), pp. 18–35.
[6] G. Astarita and G. Marrucci, *Principles of non-Newtonian fluid mechanics*, McGraw-Hill Book Company, 1974.
[7] J. Baranger and K. Najib, *Analyse numerique des ecoulements quasi-newtoniens dont la viscosite obeit a la loi puissance ou la loi de carreau*, Numerische Mathematik, 58 (1990), pp. 35–49.
[8] J. Baranger, K. Najib, and D. Sandri, *Numerical analysis of a three-fields model for a quasi-newtonian flow*, Computer Methods in Applied Mechanics and Engineering, 109 (1993), pp. 281–292.
[9] J. W. Barrett and W.B. Liu, *Quasi-norm error bounds for the finite element approximation of a non-newtonian flow*, Numerische Mathematik, 68 (1994), pp. 437–456.
[10] R.B. Bird, R.C. Armstrong, and O. Hassager, *Dynamics of Polymeric liquids*, John Wiley and Sons, 1987.
[11] D. G. Cacuci, *Sensitivity & Uncertainty Analysis, Volume 1: Theory*, CRC Press, May 2003.
[12] G. Chambon, A. Ghemmour, and D. Laigle, *Gravity-driven surges of a viscoplastic fluid: An experimental study*, Journal of Non-Newtonian Fluid Mechanics, 158 (2009), pp. 54–62.
[13] B. Christianson, *Reverse accumulation and attractive fixed points*, Optimization Methods and Software, 3 (1994), pp. 311–326.

[14] J. Colinge and J. Rappaz, *A strongly nonlinear problem arising in glaciology*, ESAIM: Mathematical Modelling and Numerical Analysis, 33 (1999), pp. 395–406.
[15] P. Coussot, *Rheometry of pastes, suspensions, and granular materials: applications in industry and environment*, Wiley-Interscience, 2005.
[16] I. Ekeland and R. Temam, *Analyse convexe et problèmes variationnels*, Bordas, 1974.
[17] A. Ern and J.L. Guermond, *Theory and Practice of Finite Elements*, Springer, 2004.
[18] V. J. Ervin, J. S. Howell, and I. Stanculescu, *A dual-mixed approximation method for a three-field model of a nonlinear generalized stokes problem*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 2886–2900.
[19] V. J. Ervin and H. Lee, *Numerical approximation of a quasi-newtonian stokes flow problem with defective boundary conditions*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 2120–2140.
[20] M. Farhloul and M. Fortin, *A new mixed finite element for the stokes and elasticity problems*, SIAM Journal on Numerical Analysis, 30 (1993), pp. 971–990.
[21] M. Filippucci, A. Tallarico, and M. Dragoni, *A three-dimensional dynamical model for channeled lava flow with nonlinear rheology*, Journal of Geophysical Research, 115 (2010), p. B05202.
[22] U. H. Fischer and G. K. C. Clarke, *Stick-slip sliding behaviour at the base of a glacier*, Annals of Glaciology, 24 (1996), pp. 390–396.
[23] J.W. Glen, *The creep of polycrystalline ice*, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 228 (1955), pp. 519–538.
[24] R. Glowinski, *Lectures on Numerical Methods for Non-Linear Variational Problems*, Springer, Jan. 2008.
[25] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting methods in non-linear mechanics*, SIAM, 1989.
[26] R. Glowinski and A. Marocco, *Sur l'approximation par éléments finis d'ordre 1 et la résolution par pénalisatio-dualité d'une classe de problèmes de dirichlet non-linéaires*, Reveu française d'Automatique, Informatique et Recherche Opérationnelle-Analyse Numérique, R-2 (1975), pp. 41–76.
[27] R. Glowinski and A. Wachs, *On the numerical simulation on viscoplastic flow*, Handbook of numerical analysis - Numerical Methods for Non-Newtonian Fluids, XVI (2011), pp. 483–717.
[28] R. Greve and H. Blatter, *Dynamics of ice sheets and glaciers*, Springer, 2009.
[29] C.D. Han, *Multiphase flow in polymer processing*, Academic Press, 1981.
[30] L. Hascoët and V. Pascual, *Tapenade 2.1 user's guide*, (2004).
[31] R. C. A. Hindmarsh, *A numerical comparison of approximations to the stokes equations used in ice sheet and glacier modeling*, Journal of Geophysical Research, 109 (2004), p. F01012.
[32] M. Honnorat, J. Marin, J. Monnier, and X. Lai, *Dassflow v1.0: a variational data assimilation software for 2D river flows*, INRIA Research Report, RR-6150 (2007).
[33] I. R. Ionescu and M. Sofonea, *Functional and numerical methods in viscoplasticity*, Oxford mathematical monographs, ISSN 0964-9174, Oxford University Press, Oxford ; New York ; Tokyo, Royaume-Uni, 1993.
[34] R. M. Iverson, *The physics of debris flows*, Reviews of Geophysics, 35 (1997), pp. 245–296.
[35] X. Lai and J. Monnier, *Assimilation of spatially distributed water levels into a shallow-water flood model. part i: Mathematical method and test case*, Journal of Hydrology, 377 (2009), pp. 1–11.
[36] B. Legrésy, A. Wendt, I. Tabacco, F. Rémy, and R. Dietrich, *Influence of tides and tidal current on mertz glacier, antarctica*, Journal of Glaciology, 50 (2004), pp. 427–435.
[37] J.L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag, 1971.
[38] H. Manouzi and M. Farhloul, *Mixed finite element analysis of a non-linear three-fields stokes model*, IMA Journal of Numerical Analysis, 21 (2001), pp. 143–164.
[39] N. Martin and J. Monnier, *Of the gradient accuracy in full-stokes ice flow model: basal slipperiness inference*, The Cryosphere Discussions, 7 (2013), pp. 3853–3897.
[40] M. Morlighem, E. Rignot, H. Seroussi, E. Larour, H. Ben Dhia, and D. Aubry, *Spatial patterns of basal drag inferred using control methods from a full-stokes and simpler models for pine island glacier, west antarctica*, Geophysical Research Letters, 37 (2010), p. L14502.
[41] C.-O. Ng and C. C. Mei, *Roll waves on a shallow layer of mud modelled as a power-law fluid*, Journal of Fluid Mechanics, 263 (1994), pp. 151–184.
[42] N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes, and O. Ghattas, *An inexact gauss-newton method for inversion of basal sliding and rheology parameters in a nonlinear stokes ice sheet model*, Journal of Glaciology, 58 (2012), pp. 889–903.
[43] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer, 2008.
[44] J. C. Robertson and R. C. Kerr, *Isothermal dynamics of channeled viscoplastic lava flows and new methods for estimating lava rheology*, Journal of Geophysical Research, 117 (2012), p. B01202.
[45] D. Sandri, *Sur l'approximation numérique des écoulements quasi-newtoniens dont la viscosité suit la loi puissance ou la loi de carreau*, Modélisation mathématique et analyse numérique, 27 (1993), pp. 131–155.
[46] ———, *A posteriori estimators for mixed finite element approximations of a fluid obeying the power law*, Computer Methods in Applied Mechanics and Engineering, 166 (1998), pp. 329–340.
[47] D. G. Vaughan and R. Arthern, *Why is it hard to predict the future of ice sheets?*, Science, 315 (2007), pp. 1503–1504.
[48] J. Weertman, *On the sliding of glaciers*, Journal of Glaciology, 3 (1957), pp. 33–38.